A Sensor Network System for Monitoring Short-Term Construction Work Zones

Manohar Bathula
Cleveland State University

A SENSOR NETWORK SYSTEM FOR MONITORING SHORT-TERM CONSTRUCTION WORK ZONES

MANOHAR BATHULA

Bachelor of Technology (B.Tech)

Electronics and Communication Engineering(E.C.E)

Jawaharlal Nehru Technological University,India

May, 2006

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

DECEMBER 2008

This thesis has been approved for the

Department of **ELECTRICAL AND COMPUTER ENGINEERING**

and the College of Graduate Studies by

Thesis Committee Chairperson, Dr. Nigamanth Sridhar

Department/Date

Dr. Chansu Yu

Department/Date

Dr. Wenbing Zhao

Department/Date

Dr. Saini Yang

Department/Date

To my mom and dad

ACKNOWLEDGMENTS

I express my sincere gratitude to my advisor Dr. Nigamanth Sridhar for his constant encouragement and inspiration.

I thank Dr. Wenbing Zhao, Dr. Saini Yang and Dr. Chansu Yu for their time as my thesis committee advisors.

I thank my department secretaries Adrienne Fox and Jan Basch for their administrative assistance.

I thank my partners Mehrdad Ramezanali, Ishu Pradan for their support in this work.

I thank all Wireless Sensor Networks(EEC693 Fall 2007) course classmates.

I thank my lab partners Madhu Mudigonda, Trisul Kanipakam, Dheeraj Reddy Bheemidi, William P.McCartney, Wook S.Kim for their useful suggestions through out the course of this work.

I thank all my friends in this country and abroad for their love and care.

A SENSOR NETWORK SYSTEM FOR MONITORING SHORT-TERM CONSTRUCTION WORK ZONES

MANOHAR BATHULA

ABSTRACT

Safety hazards encountered near construction work zones are high, both in number and in the kind. There is a need to monitor traffic in such construction zones in order to improve driver and vehicle safety.

In the past traffic monitoring systems were built with high cost equipment such as inductive plates, video cameras etc. These solutions are too cost–prohibitive and invasive to be used in the large. Wireless sensor networks provide an opportunity space that can be used to address this problem. This thesis specifically targets temporary or short-term construction work zones. We present the design and implementation of a sensor network system targeted at monitoring the flow of traffic through these temporary construction work zones. As opposed to long-term work zones which are common on highways, short-term or temporary work zones remain active for a few hours or a few days at most. As such, instrumenting temporary work zones with monitoring equipment similar to those used in long-term work zones is not practical. Yet, these temporary work zones present an important problem in terms of crashes occurring in and around them. The design for this sensornet-based system for monitoring traffic is (a) inexpensive, (b) rapidly deployable, (c) requires minimal maintenance and (d) non-invasive. In this thesis we present our experiences in building this system, and testing this system in live work zones in the Greater Cleveland area.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Construction work zones are being set up regularly in most of our neighborhoods and cities. Many construction work zones often require the motorists to share the road with the construction equipment and construction workers . Motorists driving on a roadway under normal conditions will enter an unfamiliar setting in the form of these construction work zones [1]– [3]. In 2006 1010 fatalities occurred in the work zones with 769 fatalities reportedly occurred in construction work zones, 109 fatalities in maintenance zones, 15 fatalities in utility work zones and 117 fatalities in unknown work zone types [4]. Over the past few decades accidents in such construction zones have been on the rise making them hazardous for the motorists. Crash data in these work zones have been collected over the past few years but this data was limited only to type of crash and fatalities of the crash. Transportation engineers working on improving the safety of the motorist and the worker in the work zone are primarily interested in knowing the factors affecting the accident in the work zones. These factors can be anything to deal with driver behavior, vehicle behavior or the configuration of the work zone [14]. Such factors can only be known when the traffic in these construction work zones is cautiously monitored.

Typically work zones can be classified in to two types. Long term work zones and short term work zones. Long term work zones are the work zones which remain active for few months or even few years. In contrast short term work zones will be active only for few hours or at most a few days. This kind of work zones are mostly seen in our neighborhoods.

1.1 The Problem

Over the past few years many traffic monitoring systems using inductive plates, piezo electric sensors, pneumatic tubes etc. have been built. These systems are not only expensive but are also invasive (often disrupting the traffic for their installation). The primary focus of this thesis is designing a traffic monitoring tool specifically targeted at short term work zones. Instrumenting such work zones with high cost traffic monitoring equipment already in place is not feasible. In fact in every such work zone there is no way of monitoring traffic.

Consider the following scenario. The local electric utility company needs to perform maintenance on some street for which they need to encroach into a portion of the street. The utility workers bring their equipment in a utility truck, and before beginning work, they deploy construction cones to demarcate the work area, and to warn drivers driving along this street. If there is complete blockage of one of the lanes on the street, the crew may additionally deploy flaggers to stand at either end of the work zone, to ensure that traffic is flowing in only one direction at any time. While this level of visual warning works well enough for motorists who are already driving on the street, a motorist who is a mile away, or even just around the corner, typically has no indication of the potential hazard or delay caused by this work zone. Further, in the scenario above, if a crash were to occur, the police and insurance companies are notified. From interviews of the people present at the time, and by other investigative means, the causes of the crash may be reconstructed. Such a reconstruction may be flawed: the driver in question may not divulge key errors on their part; witnesses may not have been paying complete attention, etc. Nevertheless, there is at least a record of an incident. More often than such crashes, there are cases where a motorist may come close to crashing, but is able to recover at the last instant. Such near-crashes are never recorded. These near-incidents are important, because the reason that the motorist was put in that situation may have had something to do with the design of the work zone. If there were a way of recording such instances, and studying the correlation of near-crashes with work zone design, transportation safety engineers and work zone designers could work on avoiding similar cases in the future. Our work is motivated by collaborations with the CSU University Transportation Center [10]. This UTC is specifically focused on improving safety in work zones. Through the UTC, we worked with a local flagging company, Area Wide Protective, to define the problem space and to identify the specific kinds of information that is most useful to collect from work zones.

1.1.1 Design Requirements:

Short term work zones present some unique design requirements. As short term work zones are active only for few hours, the system designed to monitor such short term work zones must be

- Rapidly deployable: The most distinct design requirement is that the system in short-term work zones need to be deployed quickly. These work zones are only active for a few hours at a time. The system, therefore, only has a few minutes to organize itself and begin producing useful data.

- Inexpensive: The total system's cost must be kept to a minimum because this is not a permanent deployment to leave the system in work zone after the work zone gets dismantled

- No skilled maintenance: Maintenance of the system must be easy and operation by the professionals cannot be assumed. Any regular maintenance activity (e.g., keeping the batteries charged) must be such that it can be performed by the construction personnel themselves.

- Self organization: No stringent requirements must be placed on the organization of the system. At the same time, though, the deployment is not totally random; simple assumptions can be made (e.g., distance between nodes will be uniform). The system can use these assumptions to aid in self-organization.

1.1.2 Data requirements:

Based on our discussions with the researchers at the CSU University Transportation Center (UTC), the most important kinds of information that needed to be collected were:

- Traffic statistics such as flow (number of vehicles per hour), density (average vehicles per mile), and average speed of vehicles traveling through the work zone. These statistics can be calculated in real-time in the work zone deployment itself. This information can be used to quantify the impact of a commissioning a work zone in that particular vicinity. This information can also be integrated with mapping software in real-time so that motorists can navigate around the work zone through an alternate route.

- Trajectories of vehicles as they travel through the workzone. The expectation is that most vehicles will maintain a straight and uniform trajectory through the work zone. When cars deviate from this uniform straight line, there is potential for crash incidents since they may come close to construction equipment or workers. Such near-crashes need to be recorded, primarily to analyze the work zone configuration to see if there was something in the configuration that may have forced the drivers to deviate from the uniform line.

- Aberrant behavior of vehicles. While vehicles may not be able to travel through a work zone at the same speed as on an unhindered stretch of road (because of reduced speed limits), the design of the work zone is intended in such a way that vehicles will still be able to maintain uniform speed. Again, cases where vehicles suddenly brake, for example, may be indicators of unsafe situations in the work zone. Our system must be able to detect such sudden changes in speed of a vehicle while moving through a work zone.

1.2 The Thesis

Increasing accidents in the construction work zones present a need to monitor traffic in these work zones. No existing traffic monitoring systems are able to meet the data and design requirements posed by the short-term work zones. A traffic monitoring system using sensor networks binding to the design and data requirements outlined in section 1.1 has been designed and implemented. Real deployment experiences using this system is presented.

1.3 The Solution Approach

Construction work zones have been hazardous over the years and are continuing to be so. There is a need to monitor traffic in such construction work zones. Throughout this research we have interacted with the transportation engineers and identified the data and design requirements for monitoring short term construction work zones. There is no system in place to monitor traffic in short term construction work zones which can meet all the data and design requirements outlined in the section 1.1. We came up with a simplest possible solution to the problem at hand using wireless sensor networks.

Over the last few years sensor networks have been used in many monitoring applications like volcano-monitoring [42], sniper localization [5] and also in traffic monitoring [19]. Wireless sensor networks also have been a natural fit for such monitoring applications.We identified that sensor networks as appropriate for the target application and focused on the key challenges in developing a sensor network system for monitoring traffic in construction work zones.

We started with identifying the appropriate sensors needed to capture the traffic statistics effectively. We identified that ranging sensors would be ideal to capture traffic statistics especially vehicle trajectories and experimented with different off-the-shelf ranging sensors. A prototype implementation of entire system has been developed for indoor testing to validate our approach. Once our hardware and software has been tested indoor to produce good results we moved to carry out some basic experiments in the CSU parking lot with higher range sensors and validated our approach. Finally we carried out experiments in Greater Cleveland Area live work zones to test our system effectiveness for monitoring live traffic. Our system collects data in work zones, and presents them for two kinds of uses: First, we provide summary information of traffic activity around the work zone for post facto analysis and research into correlating near-crash instances with work zone design, and second, we publish traffic statistics such as traffic flow, density, and average speed of vehicles for viewing over the internet. Our system publishes data in archival form to MSR Sensor map [32].

1.3.1 Contributions

1. The design and prototype implementation of a sensornet system to monitor traffic in short-term work zones.
2. Software architecture (implemented in TinyOS [17] using nesC [13]) for collecting a variety of traffic statistics, such as flow, density, vehicle trajectories, etc.
3. The use of such traffic statistics in studying the safety and efficiency of different work zone configurations.
4. Examples of real deployment experiences with temporary work zones.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows.Chapter 2 introduces wireless sensor networks. Chapter 3 presents the software and hardware design used to build the sensornet for traffic monitoring. Chapter 4 discusses the software services running on the base station. Chapter 5 presents the experimental analysis of the work. Chapter 6 presents related work section. Chapter 7 gives possible extension of this work in future work section and concludes with lessons learnt during this research.

CHAPTER II

WIRELESS SENSOR NETWORKS

2.1 Introduction

The emergence of communication, MEMS and packaging technologies have helped in miniaturizing networked embedded devices leading to emergence of Wireless Sensor Networks. Wireless sensor networks are composed of tiny, inexpensive, low power embedded networking devices which can sense physical phenomena around them using the sensors attached to them. The nodes communicate through wireless medium to form a wireless network. Some unique features of wireless sensor networks are

- Limited power and resources: Sensor networks typically operate on batteries and they have very limited resources to work with. Resource aware protocols have been developed in order to adapt to this limitations.

- Unmanned operation: Mostly these networks will be deployed in remote environments where human presence is not viable. They also must cope up with the harsh environments that they will be subjected to.

- Dynamic network topology: The topology of the network cannot be assumed prior to deployment. The network must organize itself after the deployment.

- Scalability: The vision of sensor network community is to embed the world ubiquitously with these tiny devices to get a finer control on the physical phenomenon that occur around us. Such networks comprising of tens of thousands of sensor nodes require scalable algorithms for efficient operation.

- Inexpensive Sensors: Mostly sensors used in sensornets are inexpensive and inaccurate. Inaccuracies in sensor data have to be eliminated using techniques like redundancy and filtering.

The next section describes the hardware and software features of common sensor network platforms

2.2 Hardware

The sensor network community has been working towards producing low cost sensor nodes, though most of existing platforms are only prototypes. Examples of sensor nodes are Telos [34], Mica2 [47], eyes IFX [49], Sun SPOT [48] etc. All of these hardware platforms have the following components in common on their PCBs.

- Microcontroller: It is the CPU of the sensor node. It is like a microprocessor but with limited processing capabilities and memory. Such limitations helped in reducing the form factors of the microcontrollers leading to the decrease in the overall size of the node.

- Radio: A sensor node is equipped with the radio for wireless communication. 90% of the total energy of the node is consumed by the radio for its transceiver operations.Few radios commonly used on sensor network platforms are CC1000 [51],CC2420 [50],TDA5250 [52] etc.

- Sensor: Nodes have a variety of sensors on them, capable of sensing the physical world. Some examples that other sensornet deployments have used are passive infrared, magnetic, ultrasonic, temperature and humidity sensors.

2.3 Software

TinyOS is a open source operating system for wireless embedded sensor networks. It is specifically designed to meet the operating requirements of the resource constrained embedded devices. It also comes with basic network protocols,sensor drivers, distributed algorithms etc. on which custom applications can be built [22].

2.3.1 Networked embedded systems C

NesC is a dialect of C. It is mainly designed to meet the event driven and component based architecture of TinyOS. NesC programs are made up of components. Components are wired together through interfaces. Through interfaces NesC abstracts the details of lower level components from the higher level components. Nesc has its concurrency model defined through tasks and event handlers and can detect data races at compile time [17].

2.3.2 Software Services for Sensornet applications:

Sensornet applications depending on the application criteria require certain software services to run on the nodes. Many applications require the following basic services to run the application effectively.

- Time synchronization: Many applications require all the nodes in the network agree on to a global time and remain synchronized during the course of the application. Due to cheap crystal oscillators present on most of the sensor network platforms, nodes in the network drift away from global time after getting synchronized. Software on the node negotiates the variations in the drift using handshakes [20] or internal correction techniques [21] and keeps the node synchronized to the network. Main criteria for selecting a time synchronization protocol for an application depends on the accuracy requirements and energy budget of the application.
- Localization and Organization: Sensor networks are supposed to be deployed on massive scales and careful placement of the nodes cannot be assumed. Many sensor networks must organize themselves after they have been deployed. In most of the applications, nodes must orient accurately either globally or within the network. Localization may further help in designing efficient middleware services. e.g., Spatial multiplexing [25]
- Collection and Dissemination: All the nodes in the network sense the physical phenomenon with the sensors on them and report that data back to the central base station. The network must form a routing structure for multihop wireless communication using link estimation parameters such as RSSI, LQI, ETX etc. Upon successful formation of a routing structure the data must be reliably transferred to the base station using the underlying routing structure.

  Dissemination is another network protocol with the aim of disseminating a piece of information reliably to the entire network. Dissemination protocols find use in many sensornet applications. e.g., Topology of the network has been disseminated by the base station to the nodes in network in this traffic monitoring application.

- Fault Detection: The other middleware service, needed in many sensornet applications is to be able to detect the failure of nodes. In particular, nodes need to be able to tell whether neighboring nodes are alive or not. The failure detection service provides an essential service in determining the health of neighboring nodes.

CHAPTER III

SYSTEM ARCHITECTURE AND DESIGN

3.1 Overview of the Traffic Monitoring System

Given the design requirements for the problem, we wanted to come up with the simplest design of a sensornet that would still be able to provide the appropriate kinds of data required. In order to gather traffic statistics such as flow, density, and average speed, a simple array of proximity sensors can be used to count vehicles that move past the array (similar to [43]). In order to compute vehicle trajectories, the proximity sensors would not be sufficient themselves, since the distance from the sensor to the vehicle obstruction will also be needed. Accordingly, we use an array of ranging sensors that can not only count the number of vehicles that move past the array, but also can track the trajectories that each vehicle maintains while traveling past the array. More details about the individual sensors we used are presented in section 3.2.

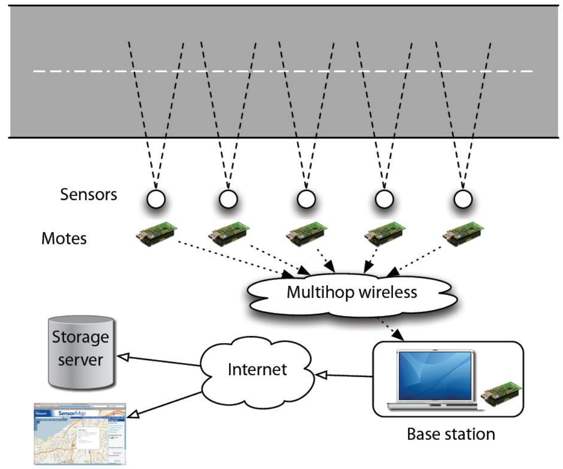


Figure 1: *Deployment architecture. Data collected from the work zone is uploaded to a server for archival and analysis and to SensorMap for display.*

A graphical representation of our network deployment architecture is presented in Figure 1 . Along the roadway of interest, an array of nodes with ranging sensors is deployed. Each sensor node is also capable of transmitting the sensed samples to a local base-station. The base-station is connected to a centralized server that is responsible for data archival and analysis. Most construction and utility trucks are equipped with a GPS receiver and a broadband internet connection, and our base-station can use this connection to access the Internet.

Sensor Placement While the sensor node placement in the network is not highly-engineered, they are placed in a predictable manner. In particular, the nodes are placed along the side of the roadway being monitored such that the following assumptions are met:

- The entire width of the roadway falls inside the sensitivity region of the sensors being used.
- Separation between nodes in the network is uniform.

This deployment architecture, and the assumptions it makes, is quite well suited for the target application. For one, the sensing hardware can be integrated easily in the work zone: they can be mounted on the safety cones used in construction zones. Further, road construction personnel in work zones already have specific parameters that they need to meet in order to put together a safe work zone. There are guidelines on distance between cones, and placement of cones. The Manual of Uniform Traffic Control Devices (MUTCD, chapter 6) describes the rules of how to place traffic control devices (safety cones, in this case) in short-term work zones [15]. These guidelines and practices can be easily exploited in the design of our deployment. The motes are mounted on top of construction safety cones. The placement of the cones is as required by the work zone itself as per the MUTCD, and is not modified by our deployment. We did not modify the construction zone, and the placement of cones in any way for this deployment. The black box (shown in inset in Figure 2) contains our sensor node hardware. The box itself is fastened to a plastic cup. This cup is placed on top of the safety cone. When mounted on the cone, the box is stable, while still being extremely simple to mount.

3.2 Hardware Design

Figure 2 shows one of our real deployment experiments. We didnt change the setting of the work zone but only used the cones already in the work zone. The black box (shown clearly in the inset) has the hardware to monitor the traffic. It contains

- *Processing and communication unit:* We use a TelosB mote [35] in the box. The USB connector on the mote is exposed outside the box, and we use this for programming and charging batteries.

Figure 2: *Our sensornet system deployed in a work zone to collect traffic statistics. The motes are mounted on top of construction safety cones. The placement of the cones is as required by the work zone itself as per the MUTCD, and is not modified by our deployment.*

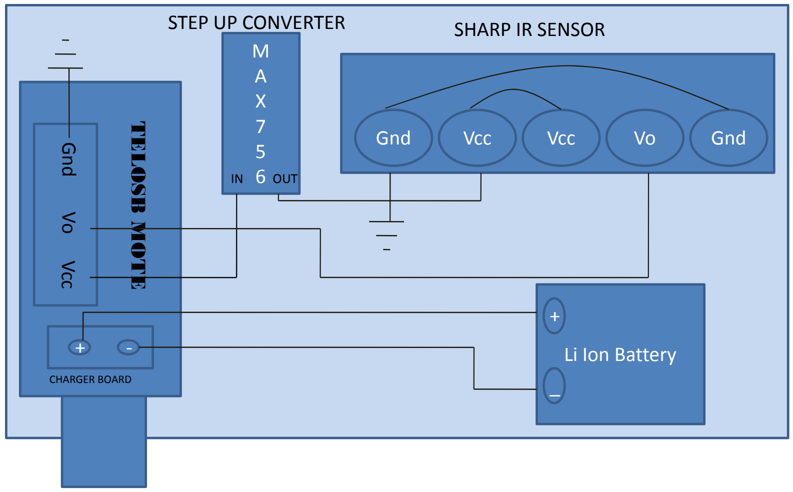


Figure 3: *Block diagram showing the internal connections of the black box*

- *Sensors:* We have used an IR sensor to detect the presence of a vehicle in the work zone. SHARP GP2Y0A700K0F IR [37]sensor was used in the deployment for the detection of vehicles. The output characteristics of the IR sensor are shown in Figure 5. This sensor has an approximate error of 1 foot in its distance reading. The sensor is connected to the Tmote Sky through the 10-pin expansion slot on the mote. A block diagram showing the connections of various components in the black box is shown in the Figure 4. Integrating the sensor to the Tmote Sky is simple.The 10-pin expansion port of the Tmote Sky is exposed so that any analog sensor can be plugged in. The *MSP430Adc12ClientAutoRVGC* component provides the necessary interfaces to expose the ADC ports of the MSP430 processor through the 10-pin expansion port. Each sensor is connected to a Tmote Sky mote, as soon as the mote senses a target, the distance reading is sent via the routing structure to the base station for processing.The operating supply voltage of this sensor is around 4.5V - 5.5V. The Tmote's 10 Pin expansion port provides a Vcc of only 3V. We used a step up converter to step up the voltage supplied by the 10 Pin expansion port to 5V. MAXIM MAX756 (the step up converter) converts voltages as low as 0.7V to 5V and operates at quiescent currents of 60μA [38].

- *TELOS Charger Board and Battery Source:*In order to simplify maintenance, and to avoid replacing batteries in the box often, we use a rechargeable battery. Further, we connect a Telos charger board [34] to the TelosB mote, and connect the rechargeable Ultralife lithium battery to the board. Whenever the TelosB mote is plugged into to a USB port, the battery is charged, and when the mote is not connected, the battery powers the mote.

3.3 Software Design

The following software services have been used on our motes:

3.3.1 Time Synchronization:

The nodes in the network must be synchronized in order to estimate the speed and flow rates of the vehicles.Time synchronization also plays a role in our localization algorithm. We used stabilizing clock synchronization protocol which provides an accuracy of 300μs. This protocol uses converge to max algorithm to achieve synchronization between the motes. Converge to max protocol is the simplest distributed algorithm for clock synchronization. Nodes periodically transmit time stamped beacons to their neighbors. After receiving the beacon the nodes will see if the received beacon timestamp is greater than its global clock, and if it is greater it will adjust its global clock to that of received beacon. The advantage with this approach is that the clocks only increase [20]. With the accuracy provided by this protocol the maxi-

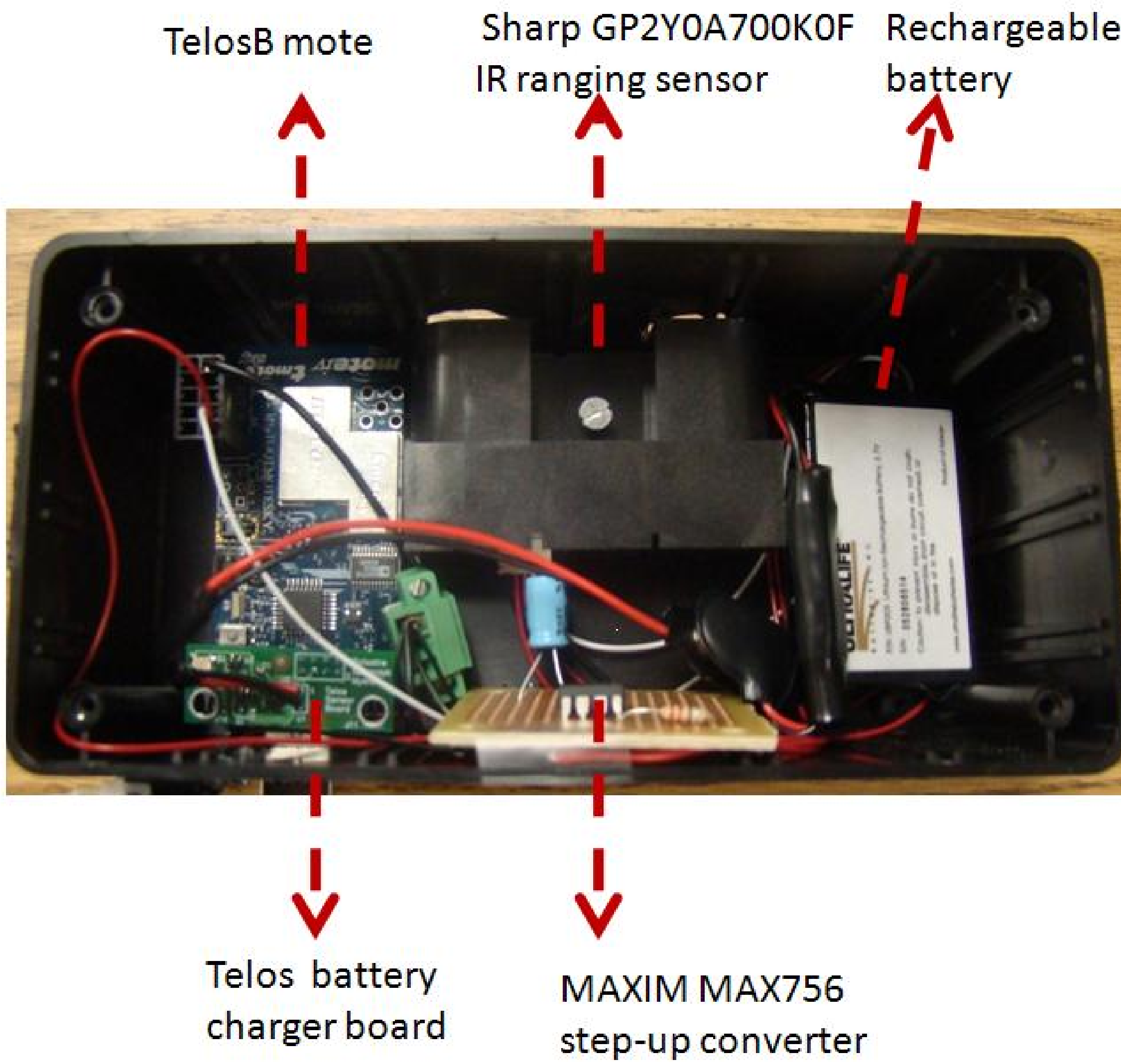


Figure 4: *Our sensor node that we deploy in the work zones. The node include a TelosB mote, with a Telos charger board connected to rechargeable batteries. The mote is connected to the Sharp IR ranging sensor*

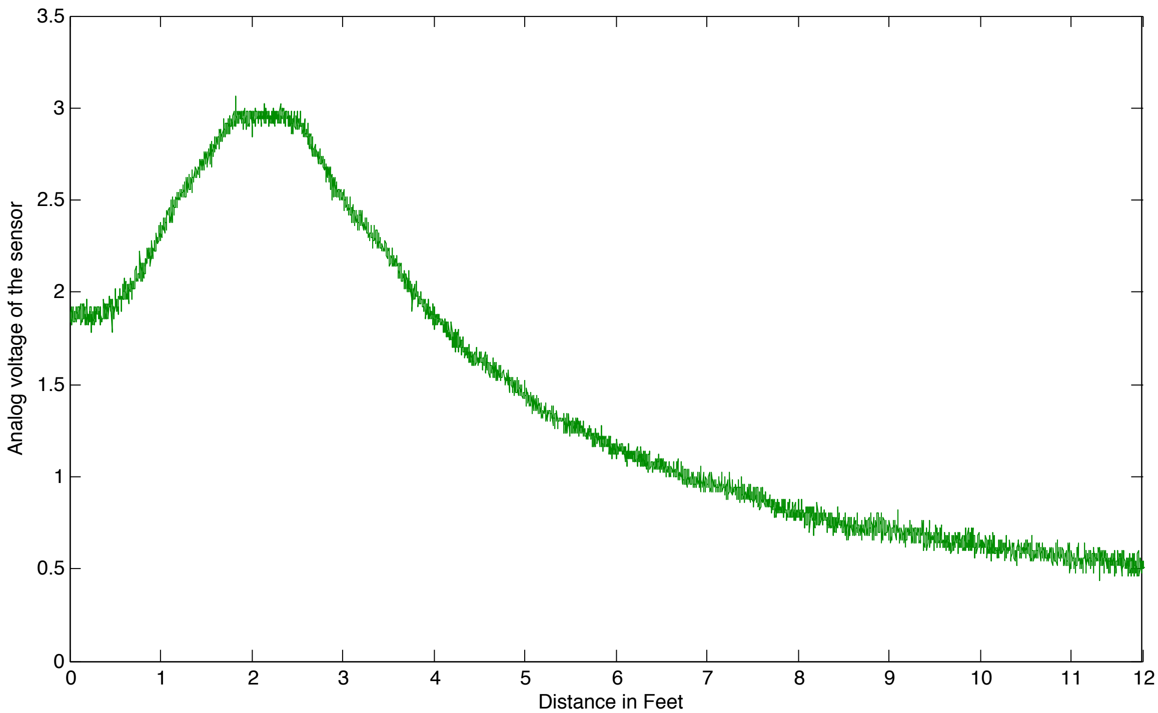


Figure 5: *Output voltage produced by the Sharp GP2Y0A700K0F infra-red ranging sensor.*

mum error introduced in the speed is about ±0.002mph. Such an error is negligible compared to speeds that will be seen in the construction zones.

3.3.2 Event Detection:

Vehicles passing through work zone must be detected and the distance of the vehicle from the sensor needed to be reported to the base station. Also the number of vehicles passing by the sensor must be counted. The vehicle counting serves two purposes.

- In estimating the flow rates of the traffic in the workzone.
- In estimating the trajectories of different vehicles later on the base station.

We distinguish one vehicle from the other using the time separation between each vehicle to pass by the sensor. The sensor is sampled at a frequency of 20 hertz. With the sensor beam width of 80cm and the sensor maximum sampling rate being 25 hertz the sensor can detect vehicles travelling at about 45 mph. With the decreased sampling rate (because of radio transmission and flash logging) it can detect cars travelling at a maximum speed of 40mph. At this speed a time gap of 200ms is more than sufficient in order to count vehicles accurately. If there is a time gap of 200ms and then if the sensor detects the presence of a vehicle then the vehicle count is incremented. The same time gap aids in finding the stop and go cars as well. The sensor finds that a vehicle has been stopped if it continuously detects the presence of vehicle without the specified time separation. So each data sample consists of three fields

Timestamp (4bytes)	Distance (1byte)	Vehicle count (2bytes)

Timestamp is the time(local time of the network) at which the sensor has detected the presence of a vehicle. Distance is the distance reading of the target from the sensor. Vehicle count is the number of cars that passed through the sensor.

Every sensed sample is logged to the external flash on each node. This is done so that post facto analysis can recover data samples missed due to lost network packets. In addition, each node keeps a growing buffer of the recent samples that have not yet been uploaded to the base-station. These samples are uploaded to the base station in batches.

3.3.3 Network Organization

When the network is deployed, before it can begin collecting and reporting data, the network has to organize itself. Note that based on the design requirements outlined in section 1.1, one cannot assume a priori node positioning. The topology of the network needs to be learned after it has been deployed. Our first attempt at network organization and localization was to use a neighbor discovery and localization algorithm based on RSSI between nodes. RSSI is the received signal strength indica-

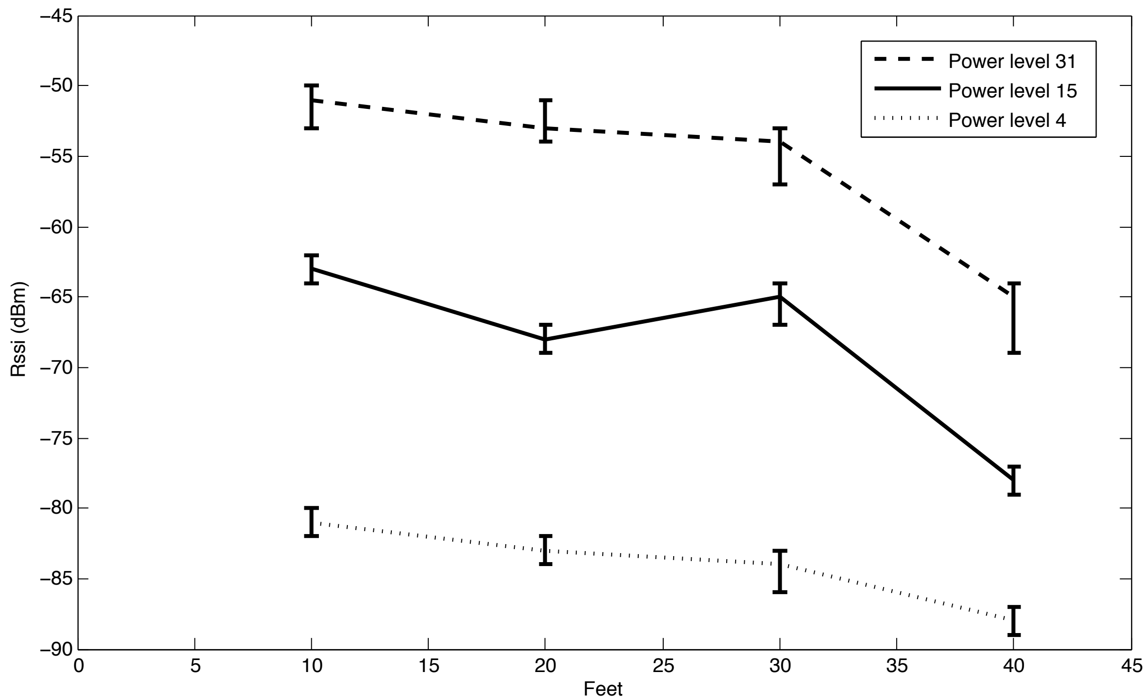


Figure 6: *RSSI at different distances. We need to distinguish between a node 10' away and a node 20' away. There is little correlation between RSSI and distance at that granularity.*

tor. Most of today's transceivers have digital RSSI and LQI support. Holland et al [18] report that RSSI is indeed a good indicator of distance. In our setup, neighboring nodes were at roughly 10 feet apart. So our primary concern is for each node to identify its two nearest neighbors (the two nodes that are 10 feet on either side), and distinguish them from nodes that are further away. The key requirement, therefore, is that a node p should be able to distinguish between a node q that is 10 away from node r that is 20 away. However, our own observations were not as consistent as [23]. In fact, we were not able to distinguish between RSSI readings at all between nodes 10 and 40 feet away (Figure 6). We were not interested in the differences beyond 40 feet since that is completely out of context in our deployment scenarios. What we observed was consistent with [40]: RSSI is good indicator of link quality at some levels, but it is not a good indicator of distance (at least at the granularity we were interested in). In [6], the authors discuss,using statistical methods or neural networks to estimate distance. We abandoned this approach since these algorithms made our system too complex, and opted for a more simple, centralized, approach to network organization: using the time-stamp information contained in the sensor messages to order the nodes at the base-station. We use a simple minimum spanning tree as the routing structure to transfer data from the network to the base-station. Once the routing tree is formed, the base-station disseminates two pieces of information to the network: (i) the depth of the routing tree, and (ii) the distance between every two adjacent sensor nodes. The base-station is provided with the inter-node distance at the time of deployment. This is the only parameter that the system needs. We keep this a deployment-time parameter because the exact physical separation between the safety cones is only known at the time of commissioning the work zone. Once the routing tree is formed, and the nodes are synchronized, they begin sampling their sensors to detect vehicle traffic. The samples are reported via multi-hop routing to the base-station, which can reconstruct vehicle paths using the time-stamp information available in the messages. Further, using the time-stamp information, the base-station can discover the topology of the network and the ordering of the nodes in the array: the time-stamps from different nodes tracking the same car will be in the order that the nodes are placed, since all nodes are synchronized. This simple sorting based on time-stamps turned out to be much more accurate than using other distributed localization schemes. We use the first 50[1] samples from each node as a training set for the base-station to converge on the topology of the network. Once the training period is complete, the base-station disseminates the topology information to the network so that each node learns its position in the array.

[1]The training set is this large to remove possible errors caused by dropped messages, missed samples, etc.

3.3.4 Data Collection

Each cycle of data acquisition needs to sample the sensors, and then report the sampled data to the base-station, if a vehicle has been detected. The motes in the network use default CSMA/CA medium access protocol provided by TinyOS 2.x [28].The CSMA/CA protocol must be able to handle the wireless traffic in the network and transfer data reliably to the base station. If all the nodes were transmitting their sensed data to the base-station at the end of every sensing cycle, the amount of data the application layer produces will be higher than what CSMA/CA can handle. This along with multihop communication will exacerbate the packet loss at different layers in the network. In our initial experiments this is exactly what we observed: the network yield, for even a small network with 9 nodes, was only around 60%.

Rather than sending a message reporting every single sensor sample, we use a delayed reporting scheme with the goal of improving goodput. The reporting scheme we use makes for spatial multiplexing in a way similar to that presented in [25]. The Flush protocol is designed for bulk data transfer over large numbers of hops. Our networks are simpler in that the number of hops to the base-station is about 4-6. We used a simplification of the spatial reuse scheme by scheduling exactly one node to upload data in each slot.

Ideally, we would like the sensing and messaging tasks to be mutually independent, such that the messaging tasks do not force a node to miss vehicle samples. Therefore, we design our batch uploads such that a node p_1 can transfer all the data it has to upload to its parent p_2 in the routing tree within the duration of time that a vehicle will take to travel from p_1 to p_2. In this manner, if p_1 begins the transfer immediately upon seeing a vehicle, then the transfer can be completed before p_2 can see the same vehicle. The default TinyOS active message payload size used in the

TelosB mote is 28 bytes [27]. In addition to the three fields above, each node will also need to include its node id (2 bytes) in each message. So each message can carry up to three vehicle samples (21 bytes). The total payload size is 23 bytes, and the size of each message is 41 bytes including header and footer sizes. If the nodes in the network are placed dnode apart, the minimum time a vehicle takes to travel this distance is Tnode, the time taken for a message to travel from sender to receiver is tm, and hmax is the maximum hop count of any node in the network to the base station, then the size of the buffer on each node is at most b:

$$b = 3 \times \frac{T_{node}}{t_m \times h_{max}} \tag{3.1}$$

In most work zones, the safety cones (and consequently, the sensor nodes) are placed 10 feet apart (dnode), and the typical speed limit is 35 mph. So Tnode is about 195ms. The message delay (tm) is about 8 ms for the 41-byte message [10], and in most of our test networks, the height of the routing tree (hmax) is 3. So the size of the buffer on each node is 24. This means that each node can cache the vehicle samples from 24 vehicles before having to transfer the data to the base-station. Moreover, in order to reduce even further the possibility of packet collisions, we employ a mutual exclusion scheme to schedule data transfers from each node. Only the node that has the mutex token transfers data, and the other nodes in the network are either idle, or are participating in multi-hop routing. The first node in the network assumes the token to begin with. When this node has accumulated b samples, it begins the transfer process. The message transfer process is started immediately upon completing a sample; this way, the sender node knows that its immediate neighbor in the array will not see the same vehicle for Tnode, by which time all the data would have been transferred. After sending all the messages (numbering b=3), it sends the mutex token to the next node in the array. The next node in the array now can begin

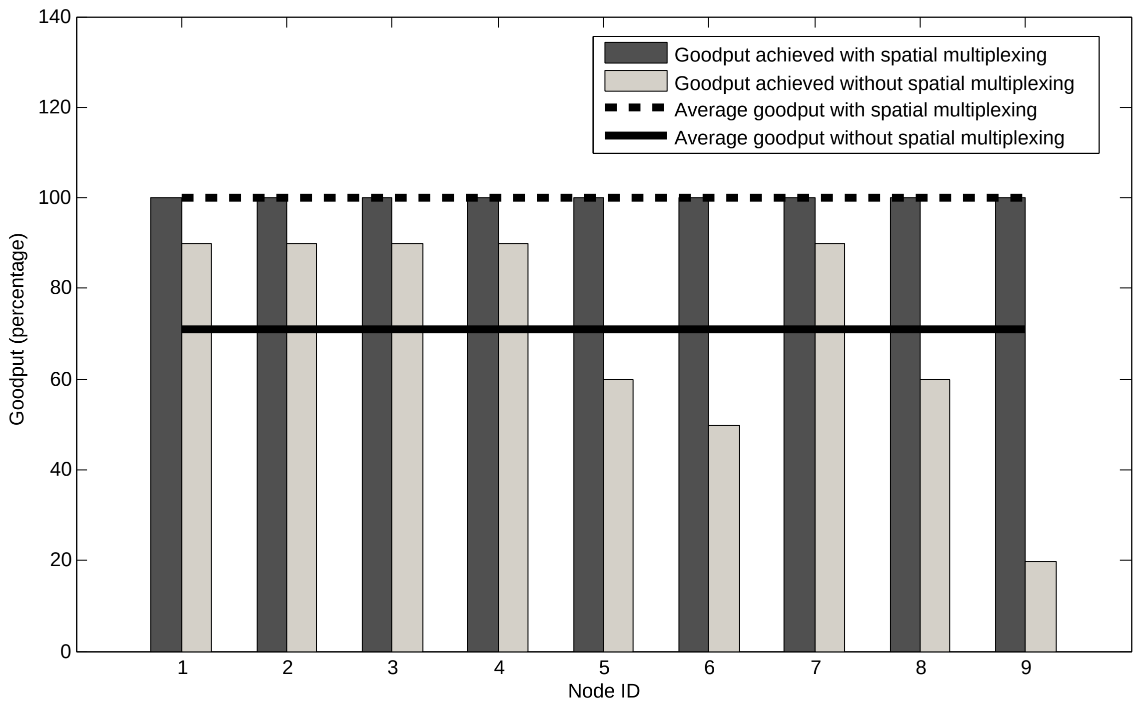


Figure 7: *Comparison of goodput with and without spatial multiplexing*

its own data transfer process. This process continues until the last node has had a chance to upload its data. Notice that all nodes in the network can transfer their cached data in the time it takes for a single car to move through the network. Once the last node in the network has transferred all of its data in that round of transfers, the base-station disseminates a completion signal. This completion signal serves to hand the mutex token back to the first node in the network, and the reporting cycle repeats approximately every b vehicles.

We compared the goodput of the network with and without spatial multiplexing(Figure 7).The spatial multiplexing technique has increased the packet yield to 100%. This has been a marked improvement in the network goodput compared to the goodput of the network without spatial multiplexing, which was around 70%.

CHAPTER IV

SOFTWARE SERVICES ON THE BASE STATION

Base station carries out three important tasks in our application

- Estimates the topology of the network and disseminates this information in the network
- Collects data from the nodes in the network and computes trajectories of the vehicles passed by the sensor array
- Publishes traffic statistics on to the Sensormap.

The topology of the network is estimated using the technique described in section 3.3.3.After collecting all the data from the nodes the base station computes the trajectories of vehicles using the following algorithm.

4.1 Computing Vehicle Trajectories

The basic idea behind our trajectory tracking system is quite simple: Whenever a vehicle crosses the sensing region of a sensor, the mote takes a sample and sends a sensor-to-target distance measurement to the base-station. This message is packaged along with the nodes ID, and its local timestamp. For now, let us consider the simplest formulation of this problem: that there is only one vehicle moving through the array.Multiple targets can be detected using the vehicle count from the mote .

A simple, naïve, implementation of trajectory mapping can simply take these points, and order them in time and space. Our topology is learned by the base-station during the training period. Using this information, the base-station can locate a node with ID k to a particular (x_k, y_k) coordinate location. This coordinate location, in addition to the target distance, can be used to compute the targets coordinate location at time t_k — $(x^{t_k}_{target}, y^{t_k}_{target})$. With sensor readings from all the nodes in the array, the base-station can assemble an ordered list of points through which the target traveled. A simple curve passing through these points will give us an approximation of the actual path the vehicle took. In fact, if we had a very high amount of confidence in the sample point readings that the sensors returned, this approach will likely be good enough to provide usable data. Unfortunately, however, our sensors are not as accurate. The sensor, based on our calibration, has an error margin of about a foot. This is nearly 7% of the entire sensing range! Such a high error margin is hardly useful when dealing with a problem as important to daily life as work zone safety.

To improve the accuracy of the trajectory mapping algorithm, we implemented a particle filter algorithm based on the one in [39].

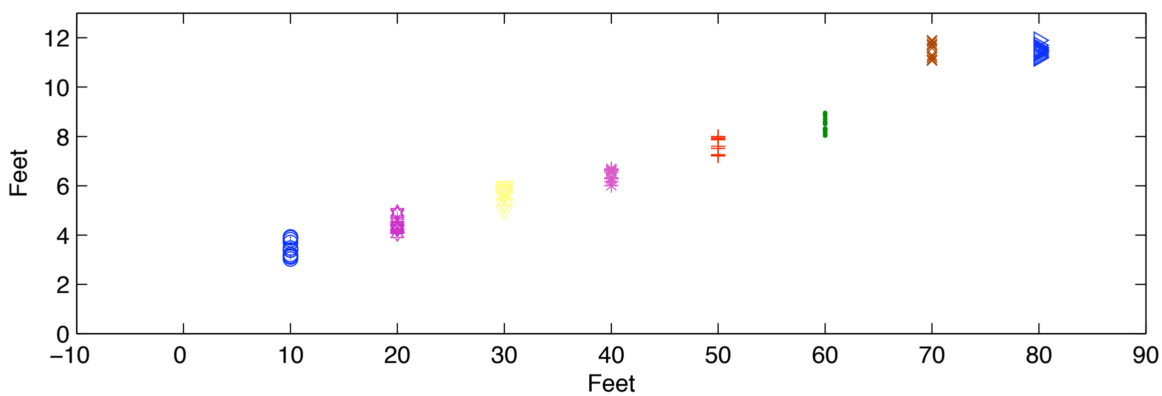
4.1.1 One-Dimensional Particle Filtering

We use this centralized algorithm to reduce the extent of dependency on the sensor and its calibration alone for trajectory mapping. In the formulation in [39], the authors use a particle filter in order to detect multiple targets in a 1-dimensional space using binary proximity sensors. Our space is a 2-dimensional space, and we modified the setting accordingly. Our use of the 1-dimensional particle filter is to basically get a more accurate reading of the distance (more accurate than the error margin of the sensor itself would allow). In our 2-dimensional space, the sensors are arranged along the x-axis, and let us now suppose that the sensors range is a straight line along the y-axis. Just like before, each sensor sends its timestamped distance-to-target reading along with its ID. At the base-station, for each sensor k, the particle filter algorithm generates a set of n particles P_{k1}, P_{k2}, ..., P_{kn} along the y-axis, where each particle is a coordinate location with the same x-coordinate as the sensor's location, with just the y-coordinate varying $(x_k, y_{P_{k1}})$, $(x_k, y_{P_{k2}})$, ..., $(x_k, y_{P_{kn}})$. Figure 8(a) shows the particle patches (in the y-dimension) that are used in finding candidate trajectories.

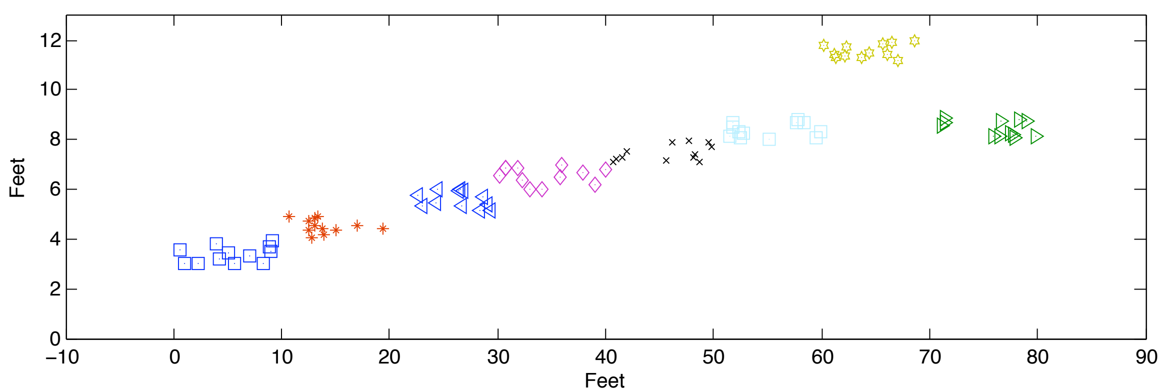
This set of n particles is generated for each of the m sensors in the array. Particles P_{11}, P_{21}, ..., P_{m1} form trajectory candidate, TC_1. Similarly, particles P_{12}, P_{22}, ..., P_{m2} form TC_2, and so on. Once all the n trajectory candidates have been calculated, a cost function is applied to select the best out of this lot. The cost function we use is also simple: trajectory candidates that have wildly abrupt changes in velocity or trajectory will have a high cost. The trajectory with the least cost is selected as the one that will be used.

4.1.2 Two-Dimensional Particle Filtering

While the 1-dimensional particle filter presented above improves the precision of our trajectory mapping system, there is still room for improvement. To further



(a) Particle patch in one(y) dimension



(b) Particle patch in two dimensions

Figure 8: *Particle patches*

increase precision, we extended the particle filter to two dimensions. Rather than generating particles in only the y-axis and keeping the x-coordinate constant, the 2-dimensional particle filter generates $n$particles $(x_{P_{k1}}, y_{P_{k1}})$, $(x_{P_{k2}}, y_{P_{k2}})$, $\ldots$, $(x_{P_{kn}}, y_{P_{kn}})$ in 2-d space. Figure 8(b) shows the particle patches (in two dimensions) that are used in finding candidate trajectories.

The rest of the trajectory mapping algorithm is the same as above. A set of n trajectory candidates is calculated, and based on the cost function, one of them is selected as the best-fit trajectory.

4.2 Detecting Aberrant Behavior

Sudden changes in speed of vehicles typically indicate potentially unsafe physical situations on the roadway. Estimating the speed of a passing vehicle is quite simple with an array of sensors. Using two time-stamped observations of a vehicle, and the distance between the two sensors, the speed of the vehicle can be computed. By calculating speed between every pair of sensor nodes in the array, we can get the speed of the moving vehicle in different regions of the work zone. In the normal case, one would observe a uniform speed, or a gradual increase or decrease of speed. Sudden fluctuations of speed (over 10% change with 20 feet, for example) are triggers to flag a vehicle as moving in an aberrant fashion. The number of such instances are recorded, along with the region of the work zone where the sudden speed change occurred. By examining this data post facto, deductions can be made about potential safety hazards in the design of a work zone.

4.3 Publishing Data for Wide Access

On a typical day, there are tens, even hundreds of short-term work zones that are active. Our partner, Area Wide Protective, alone deploys a number of active work zones in the Greater Cleveland area. One of the biggest problems with short-term work zones is that there is typically no record of its existence. In fact, except for motorists that are driving along the street on which the work zone is commissioned, no one even knows about it, unless the work zone is causing such a large impact on traffic that the local radio station were to include it in its traffic broadcast; this might be one out of fifty active work zones.

Most work zones, therefore, go unreported. And it is quite likely that a large majority of commuter motorists will come across at least one active work zone (that they did not expect to see) on their way to work. While traffic information on major highways in metro areas is already available in mapping services such as Google Maps [18] and Microsoft Live Maps [29], traffic delays caused by short-term work zones are not reported.

In our system, the base-station uploads synthesized traffic data to the internet. Most construction utility trucks have a live internet connection, used for other maintenance and monitoring purposes. Our system can upload sensor data, currently to Micosoft's SensorMap [32][1]. The information that we make available are: (*i*) traffic flow and density, (*ii*) average speed of vehicles. After the work zone is taken down, we leave the data archived in the SensorMap database for a period of time. When our systems are ready for wide deployment, we will also have integration with mapping systems to provide real time data about short-term work zones in driving directions.

[1]The interested reader can search SensorMap V3 for zip code 44114 to find our recent uploads.Click on the time traveler and adjust the resolution to minutes

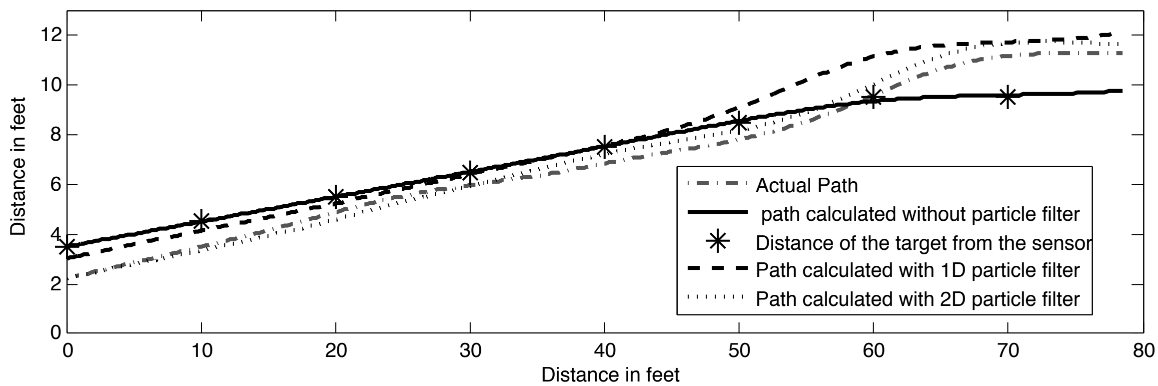
CHAPTER V

EVALUATION AND RESULTS

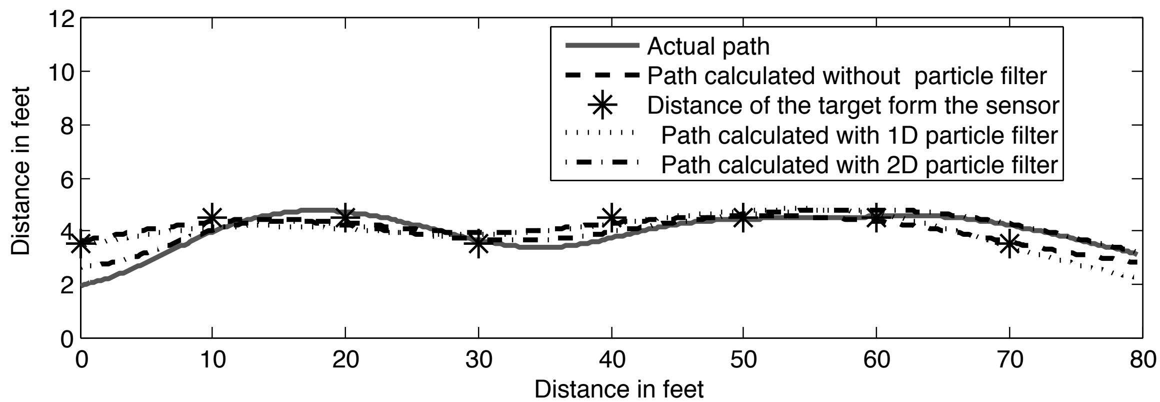
5.1 Estimating Vehicle Trajectories

Experiment Setup We tested our trajectory mapping algorithms in a testbed deployed in a parking lot with eight nodes with IR sensors. The nodes were placed ten feet apart from each other, in a straight line. One of the motes acted as the root of the collection routing structure, and communicated with a PC acting as the base-station. We drove a car in a pre-determined path as our target moving through the sensor array. Each sensor took 10 samples/sec. This sampling rate was sufficient to capture the target moving through the array, based upon the speed of the target moving across the array, and inter-node distance.

We ran experiments with two different paths. The results are shown in Figure 5.1. In the case of each of the two paths, four curves are shown. One of these is the actual path traveled by the target vehicle. The first calculated curve simply takes the sensor readings, directly. These readings, based on our sensor calibration tests, may be off by up to three inches from the actual path.



(a) Path1



(b) Path2

Figure 9: *Comparing the actual path of a target across the sensor array, and the trajectory we computed. We drove our target along two different paths, and tested the three versions of trajectory mapping to approximate the target's path.*

The second calculated curve uses 1-d particle filter (along the y-axis) to better approximate the reading, and to compensate for sensor calibration error margins. The final calculated curve is the curve calculated using the 2-d particle filter. As one can see from all the three different paths we tested with, the accuracy of the computed path becomes better as we move from plain sensor calibration, to 1-d particle filtering, and finally to 2-d particle filtering.

Accuracy of Particle Filter We also measured the effect of the number of particles generated. For different numbers of particles in the particle filter, we measured the error rate (characterized by the average distance of the calculated curve from the

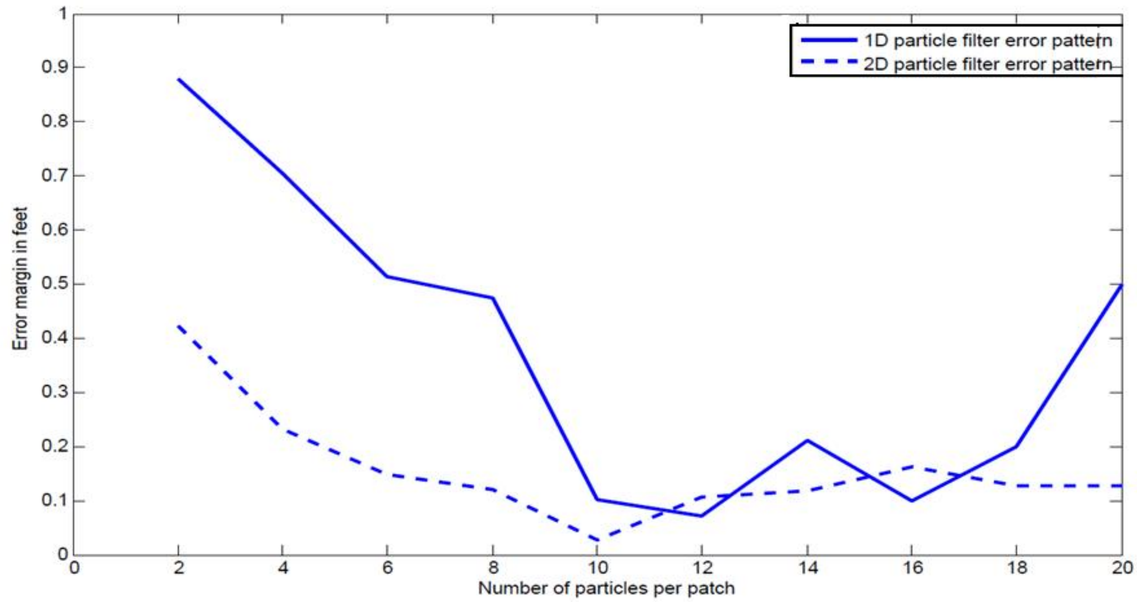


Figure 10: *Error margin with number of particles in patch*

actual path). Figure 10 shows the error rate as a function of the number of particles for the 1-d as well as the 2-d particle filters, and the 2-d filter indeed performs better.

5.2 Deployment Experiences

We deployed our sensornet system on work zones deployed by Area Wire Protective (AWP), a flagging company in Northeast Ohio. The company provides road work zone services to a number of utility companies in the area. When a utility company (gas, electric, cable) has to perform maintenance work that may cause traffic restrictions, AWP sets up a work zone for them to ensure safe operation.

In this section, we report data we collected from one of these work zones in the Greater Cleveland area[1]. The location of the work zone was on Lorain Road near the intersection with Clague Road in North Olmsted. This is a pretty busy road, and in one hour during our deployment, we observed 614 cars pass through the work zone.

[1]The data collected from two other test deployments are similar in kind. The complete collection of datasets is available at http://selab.csuohio.edu/dsnrg.

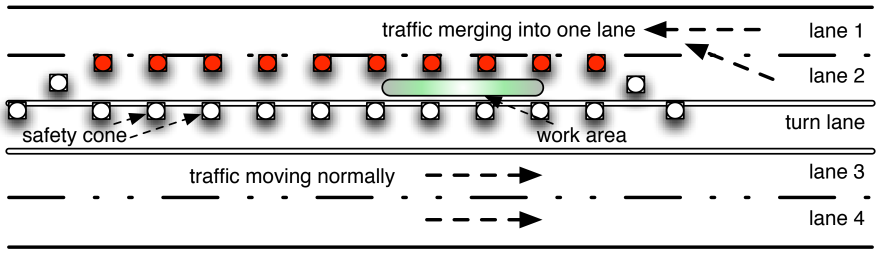


Figure 11: *Configuration of the work zone used in the case study. Our sensor nodes were placed on the shaded safety cones to monitor traffic along lane 1.*

We videotaped the traffic during the entire deployment in order to serve as ground truth to compare against our sensed measurements.

The work zone that we discuss was about one hundred feet long, and occupied one lane of the street. The actual work area was in the middle of one of the drive lanes, and was about 20 feet long. The street had two drive lanes in either direction, and a turn lane in the middle (shown in Figure 11). The work zone guided the traffic to merge from two lanes into one. We deployed our sensors to monitor traffic flowing along the lane that carried the merging traffic (lane 1).

Average speed of vehicles The speed limit on Lorain Road is 35 mph, and there was no reduction in speed limit caused by the work zone. This is also typical – short-term work zones rarely cause speed limits on streets to be reduced, unless there is complete blockage of one direction of traffic. There was a traffic light about 500 feet downstream from our work zone, and this caused some slowdowns and some stopped traffic as well. The average speed of vehicles driving through our work zone was about 12 mph. Further, we measured speeds between every pair of nodes, *i.e.,* average speed in every 10-foot segment in the work zone. These speeds are shown in Figure 12. Notice how the average speed of vehicles is lower immediately upon entry into the work zone, and just before exiting the work zone. Near the middle of the work zone, motorists generally tend to be “more confident,” and hence tend to speed

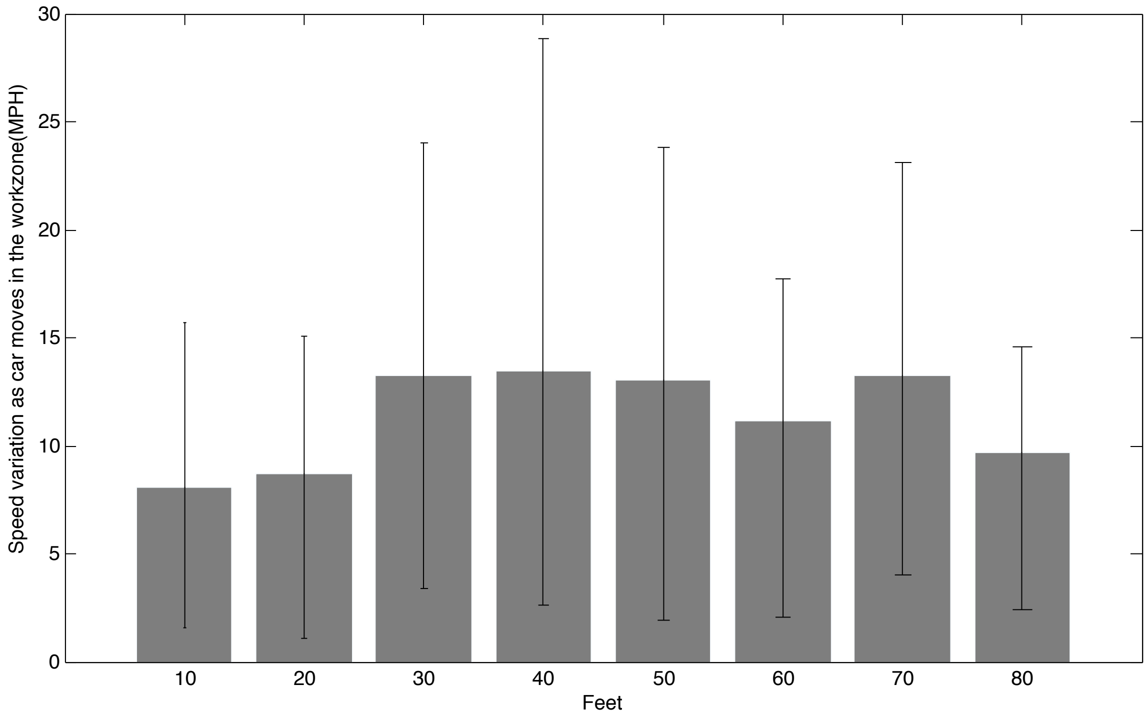


Figure 12: *Average vehicle speeds in different segments*

up a little. In spite of the speed limit being 35 mph, we didn't actually observe any vehicles traveling as fast. This was mostly because of the density of traffic, which was "bumper-to-bumper" for most of the time the work zone was active.

As another measure of how fast vehicles are moving through the work zone, we show cumulative time-location plots of vehicles in Figure 13. Looking at this figure, we can see that most cars spend about 10–20 seconds in the work zone, while a small number of them spend longer.

Changes in speed Sudden changes in vehicle speeds is another point of interest for work zone designers. If a number of vehicles either sped up, or suddenly braked at a particular spot in the work zone, that spot merits some special consideration. Figure 14 shows the number of cars that changed speed by over 5 mph in adjacent 10 ft segments. See the correlation between this graph, and the graph in Figure 12: a number of cars speed up in the second segment, resulting in a higher average speed

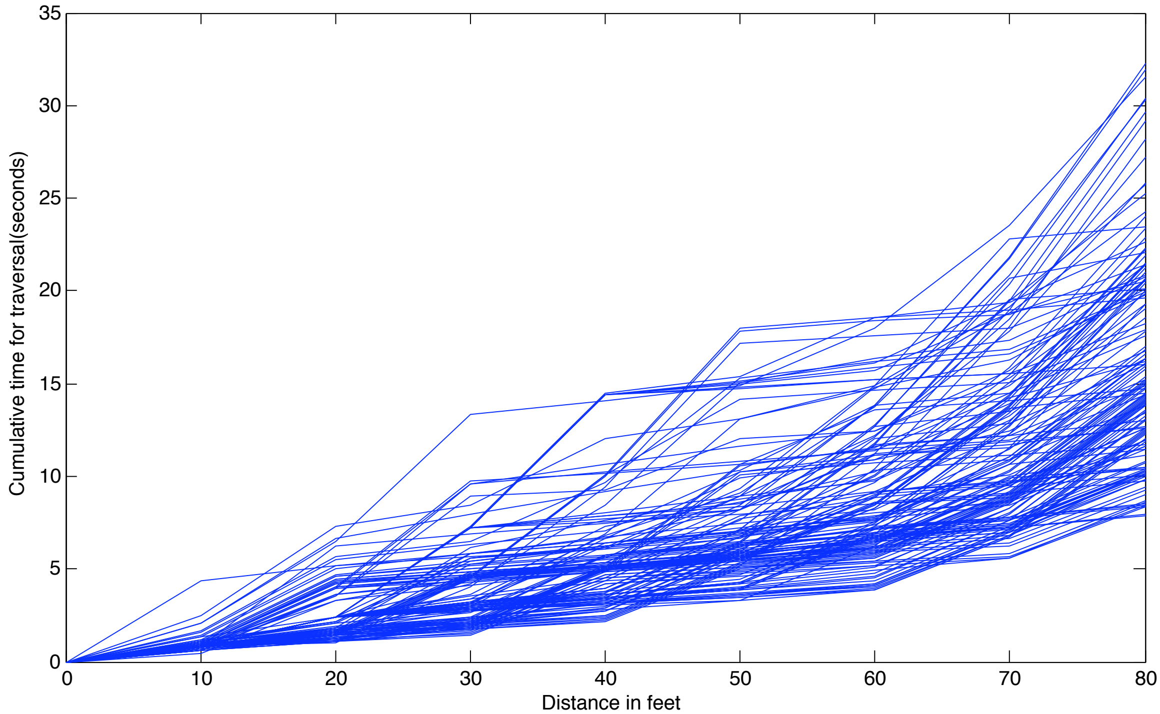


Figure 13: *Cumulative time of traversal for vehicles driving across the work zone*

in the middle of the work zone. Again, near the end of the work zone, a number of cars reduce speed just before exit.

Rate of flow of traffic Figure 15 shows the rate of flow of traffic during the hour of data capture. As we can see here, for most of the time, the work zone had a fair number of cars driving through it. There are very short intervals of time when the flow rate was less than 5 cars per minute. This is a good way for us to validate our sensor sampling rate. Even in dense traffic, our sensornet is able to produce good data. As we said earlier, we videotaped the traffic during this time, and compared it with the data collected from the sensornet. We found the data to be very well correlated with the video data.

Vehicle trajectories Using the trajectory mapping scheme described in Section 4.1, we calculate trajectories of the vehicles driving past our sensor array. During our de-

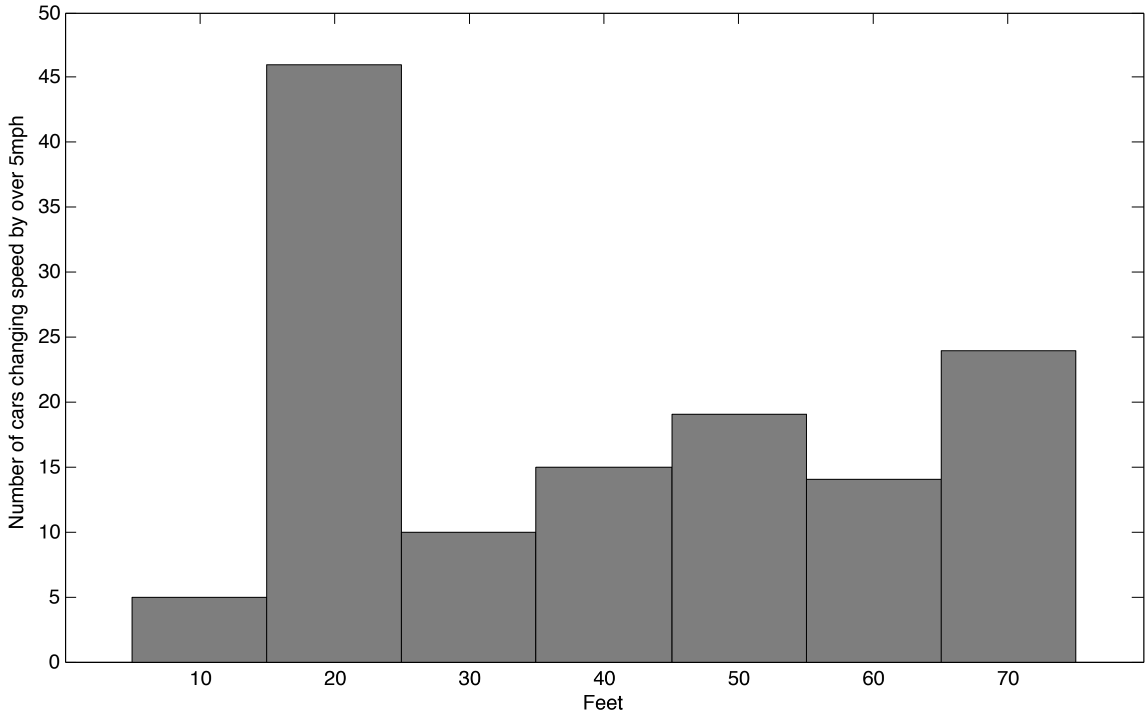


Figure 14: *Number of cars that changed speed by over 5 mph in adjacent 10 ft segments*

ployment case study, we observed a majority of vehicles maintaining a steady path through the work zone. The average trajectory was about 4 feet from the side of the lane (Figure 16). The width of the lane is 12 feet, and the average car is about 6 feet wide. Given this, if cars were driving perfectly in the middle of a lane, then they would be 3 feet from either edge of the lane. The tendency of most drivers, when they see safety cones or other construction equipment, is to tend away from them, and favor driving closer to the opposite edge of the lane. This anecdotal tendency is confirmed in our case study instance, where the average trajectory is a foot further than "dead-center" of the lane away from the safety cones.

However, a number of cars did veer off the average trajectory, and some came too close to the safety cones, and some others were driving too close to the opposite curb. Figure 18 shows the number of cars that veered too close to the safety cones measured at each of the sensor nodes. These instances are of interest to work zone

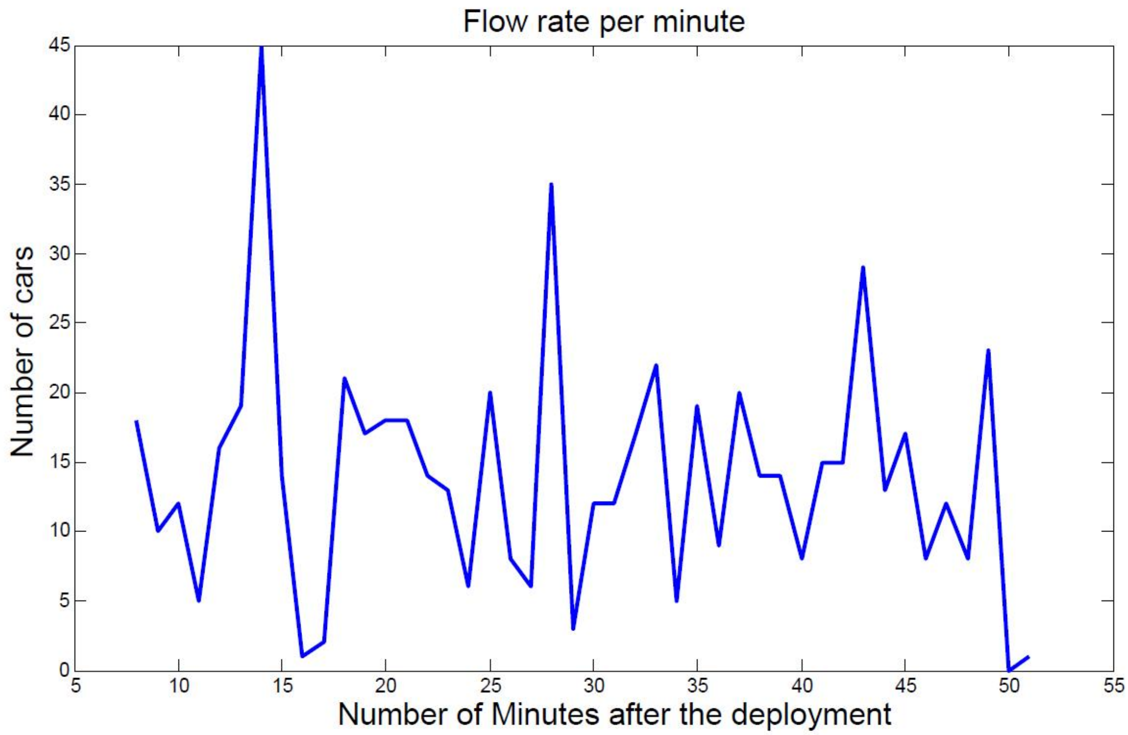


Figure 15: *Flow rate of the traffic in the work zone*

designers: if there were an inordinate number of vehicles leaving the preferred trajectory at any particular part of the work zone, that may indicate a potential unsafe situation. In our case, there is no such unusual observation, indicating that the traffic in this work zone was mostly "docile".

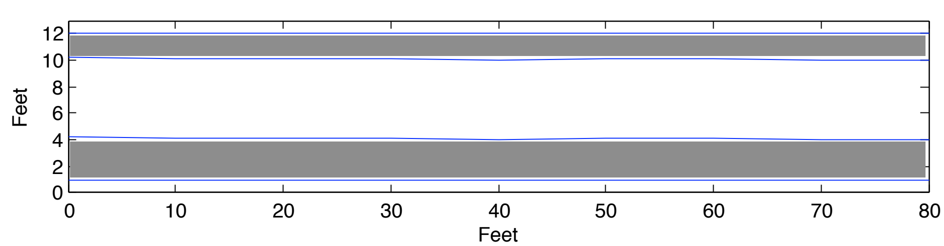


Figure 16: *The white region in the middle denotes the average trajectory of cars. The shaded regions on either side denotes the extent of deviation from the average (safe) trajectory.*

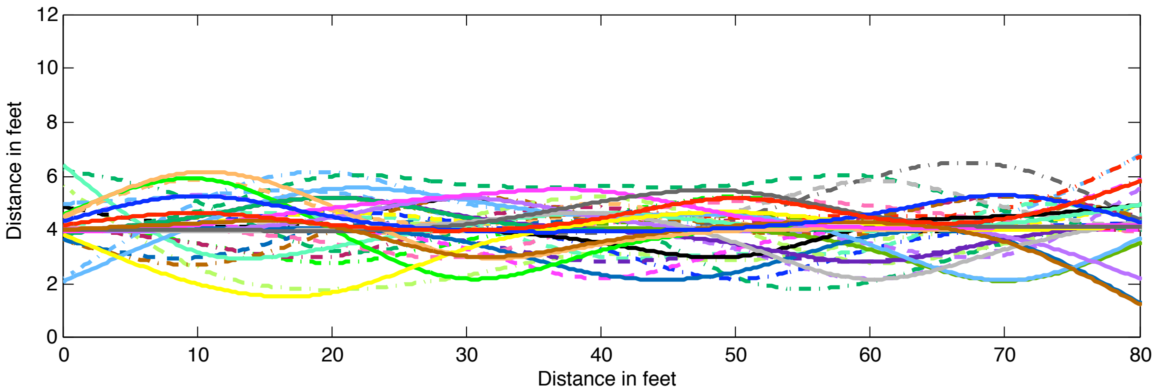


Figure 17: *The trajectories of 50 cars out of our total set of 614. Note that this is just the side of the cars nearest to the sensors.*

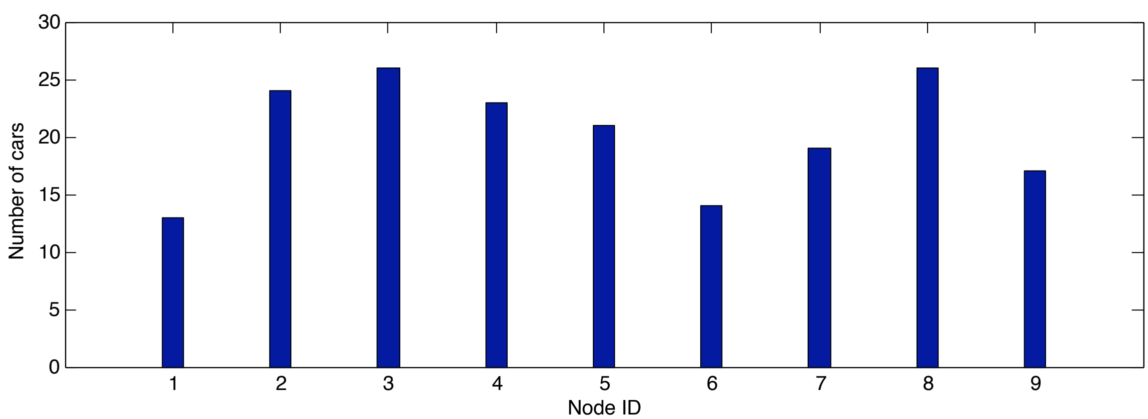


Figure 18: *Number of cars that went off the average trajectory as detected by each node in the work zone*

CHAPTER VI

RELATED WORK

Current traffic monitoring systems can be classified as invasive,non invasive and off-road systems [45]. Inductive loops, pneumatic tubes, piezoelectric sensors, Weigh-In-Motion (WIM) systems are some examples of invasive systems. The deployment of such systems requires activities such as digging the road, closing the path for some time etc. often disrupting the traffic. Moreover these devices are very expensive. Non invasive systems don't require any installation. They can be placed on the street lights, pavements etc. to monitor the traffic without affecting the flow of the traffic. Sensors like magnetic, infrared, ultrasonic, video cameras were used in such deployments. Video cameras can provide richer resolution data compared to other sensing modalities but require large amount of bandwidths, storage spaces and high mounting points. Merely a set of two sensors from above mentioned non-invasive sensors can detect the presence of a vehicle and estimate the density, speed and flow rates. In order to find the near crash scenarios, trajectory of the vehicle is an important statistic. Trajectories can only be computed with array of such sensors(exception – video cameras).Off road systems use devices like GPS receivers,mobile phones,PDAs used by the general public to monitor the traffic in the areas where such devices are available.Such systems do not require any installations on the road for monitoring traffic.Concerns with this approach are the availability and privacy of the user.

In our system sensor nodes can be deployed with minimal engineering efforts (at most placing them on either side of road or along a straight line on one side of the road with some precise separation). And compared to deployments involving inductive loops, measuring poles etc., sensornets come at a substantially lower cost. Indeed others have used sensornets in the traffic monitoring context. In [7] and [21] authors have deployed sensors in the intersections of freeways and parking lots. [15] describes wireless magnetic sensors that can be used for traffic classification and surveillance. The sensors are designed to identify vehicles, speed of the vehicles, conditions of the road, density of the traffic etc. But the problem we deal with is, apart from monitoring the traffic, we are interested in the trajectories of the vehicles especially in work zones so that the traffic authorities can now learn about near-crashes which are impossible to find. Work zone designers can use this information in designing work areas that provide a safer environment to both drivers and workers. There are also other mechanisms which directly deal with the driver rather than the vehicle [25]. These systems simulate traffic and may be subject to errors. In [37], Yoon et al. show how to estimate traffic on streets using GPS traces. In their system, cars are equipped with GPS receivers, and the traces of these GPS receivers is used to analyze traffic patterns. Using their system as well, they are able to show that road work and work zones do have an impact on traffic patterns. The CarTel system [19] also equips cars with additional sensors that can actually hook into the cars electronics, and therefore acquire information about the vehicles internal statistics, such as speed, engine RPM, etc. By tagging these data samples with GPS locations, they are able to produce rich traces of information captured during drives. For example, they can statistically compare a number of different routes between two points in terms of travel distance, and pick the best one. The Nericell [24] system is similar in that they use sensors in moving vehicles. As opposed to [37] and [19], however, their focus is on using sensors that people carry with them anyway. Their work is focused on using smart cell phones (which have a number of sensors such as GPS, microphone, accelerometer, etc.) to derive vehicle traces. By using this heterogeneous sample of traces, they can identify potholes on roads, distinguish traffic stopped at a red light from traffic stopped in a jam, etc. All these systems are complementary to our work, since they involve embedding sensors in moving vehicles. One of the most comprehensive, publicly accessible, systems for traffic monitoring is the Intelligent Transportation Systems (ITS) division of the California Department of Transportation (Caltrans) [5]. In addition to live feeds from a number of sensors across the state of California, the website also provides a wealth of information in the form of studies and reports focused on monitoring traffic. The Ohio Department of Transportation (ODOT) maintains a similarly rich webaccessible system called Buckeye Traffic [27]. The ODOT website maintains and provides current information about road closures and restrictions on major highways because of construction projects, and identifies road activity from a variety of permanent sensors all over the state of Ohio. While this system is comprehensive in capturing road activity on major highways, no short-term work zones are captured; it is not economically feasible to have permanent sensors deployed in every street. The same is true also of the Federal Highway Administrations (FHWA) ITS initiatives [12].

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 Future Work

We are yet to consider some more important aspects if this technology is to find wide applicability. A primary consideration is to weather-proof the node, not just from a physical standpoint, but also from a functional standpoint. The infra-red sensor requires that there be holes drilled on the physical enclosure of the node. These holes present two problems first, the node is no longer safe against the elements, and second, the lenses on the sensor are exposed to dust and other particles that may cause distortions. In order to get around this problem, we are currently engaged in researching other sensors that can provide similar ranging capabilities while still being inexpensive and easy to use. As of this writing, we are experimenting with the SRF02 ultrasonic range finder [8]. This sensor has a similar range as our Sharp infrared range finder (6 m). The sensor can connect to the TelosB mote through the I^2C interface, and directly provides a distance reading (in cm) based the obstruction in front of the sensor. The sampling time of this sensor, however, is twice that of

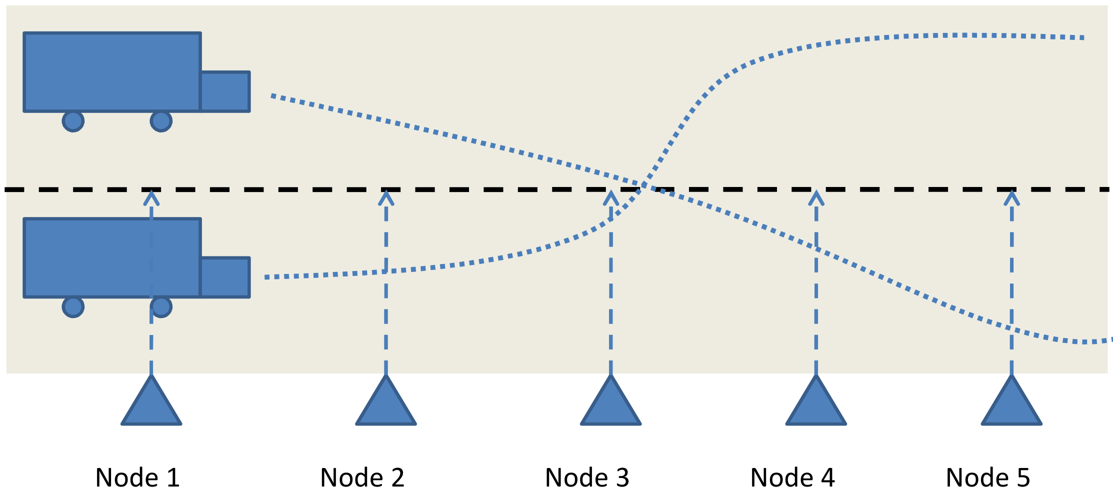


Figure 19: *Graphical representation illustrating the problem with multi lane vehicle tracking*

the IR sensor, which may cause timing issues with respect to capturing traffic: this sensor may miss some vehicles because of the reduced sampling rate.

The other important aspect that needed to be considered for this application to find wider applicability is to extend the tracking system to detect vehicle trajectories in multiple lanes. The key challenge in finding vehicle trajectories in multiple lanes is to detect vehicles changing lanes randomly. Consider the following scenario(Figure 19) Say, two vehicles were travelling on a two lane road, each in a separate lane. For simplicity assume that nodes were placed only on one side of right lane of the road. Initially Node 1 detects the car in the rightlane and counts it as the first vehicle. If the car in the left lane decides to move to the right lane after the first car has been detected by the node1, the nodes 4 and 5 will detect and count the car from the left lane (moving into to the right lane) as the first car. If the car in the right lane also moves out of the lane before reaching nodes 4 and 5 the vehicle count of the entire network after these two cars had left the sensing region would be one. Simple curve passing through the distance readings of the car count one will only give a wrong trajectory. An approach like the one based on [46] can be taken to get around this problem. Nodes should exchange information with the network about the possible arrival of vehicles that are about to change lanes.The potential change of lane by the car can be known using the distance reading of the sensor.

7.1.1 Other applications

Though this system has been specifically designed for monitoring short-term work zones it can find applicability in other applications related to parking and traffic management. This system can be used for parking lot space management. The number of cars coming in and out of the lot can be counted and the spaces available in the lot can be displayed in the entrance of the lot. Another potential application could be,using this system at traffic lights.Traffic lights at intersections can be controlled based on the density of the traffic, rather than using simple timers.Such systems already exist in most of the cities but most of them make use of inductive loops.

7.2 Conclusion

Throughout this research, during design and development, we constantly worked along with transportation engineers from the CSU UTC to guide our work along, and to make sure that we meet our design guidelines in the best way possible. Comparing with the design requirements we set up in Section 3, our system satisfies all of them quite nicely:

1. The nodes are pre-programmed to collect traffic statistics, and the base-station is programmed to synthesize this data for publication on the internet. Deployment of the network simply entails mounting the nodes onto safety cones and

turning them on.

2. Our prototype node is built from off-the-shelf parts, and as such, the cost of all parts in the node add up to about $220. We expect to cut this cost in about half when we switch these parts with mass-fabricated custom parts. A cost of $100 per node is still within the "inexpensive" limit, according to our partners. In comparison, most sensor equipment deployed in long-term work zones run thousands of dollars per node.
3. The only regular maintenance that is needed for our sensor nodes is to keep the battery charged. We have conveniently exposed the USB connector of the TelosB mote for this purpose. Simply plugging in the mote will charge up the battery. This is an activity that the construction company can do every night, ensuring that the nodes are ready for use in the morning.
4. Our network does not expect to be deployed in a pre-determined fashion. Instead, we built our self-organization logic around the practices that the construction work zones follow. Accordingly, we know that the nodes will be placed at uniform distances apart from each other. Moreover, we exposed the only parameter that will need to change – the inter-node distance. This parameter is disseminated to the nodes by the base-station upon startup.

While we were working on writing this thesis, we came across advice for successful sensornet deployments [3], based on a variety of different deployments of the SensorScope project [4]. While clearly the advice contained in this paper would have been extremely useful earlier on in our research, we were quite pleasantly surprised that we had already followed a number of the good practices [3] listed. For example, from the beginning, we have worked with domain specialists from transportation engineering to define the problem, and to expose the solution spaces; we have always trusted experimentation with real hardware as opposed to simulation (we used a smaller version of the IR sensor for lab tests with toy cars early on); our protocols are as simple as they can be, making system behavior very predictable.

Construction work zones on roadways are hazardous areas, primarily because they cause an otherwise familiar setting to become unfamiliar. Motorists driving through a roadway under construction may end up facing unexpected scenarios. Long-term work zones on major highways may present such unfamiliarity in the beginning, but once a motorist has driven on the modified road a few times, she can get used to the changes (which will last a few weeks, if not months). By contrast, short-term work zones the kind that we see in our local city streets for utility work are only active for a few hours at a time. This transient nature leaves them untraceable for the most part. In fact, there is very little empirical data available about traffic scenarios in short-term work zones. In this paper, we have presented the design and prototype implementation of a sensornet system that is specifically targeted at collecting data about traffic in and around short-term work zones. Our system is rapidly deployable, easily maintainable, and is capable of capturing a variety of different statistics about vehicle traffic in work zones. The data collected can be used by transportation engineers to consider design parameters for future work zone configurations. We have tested our systems in live work zones in the Greater Cleveland area, and are now in the process of working on expanding to wide deployment.

BIBLIOGRAPHY

[1] J. Wang et al, "Investigation of highway work zone crashes: What we know and what we dont know ", Transportation Research Record, vol. 1529, pp. 5462, 2007.

[2] J. R. Treat, N. S. Tumbas, S. T. McDonald, D. Shinar, R. D. Hume,R. E. Mayer, R. L. Stansifer, and N. J. Castillan, "Tri-level study of the causes of traffic accidents ", Indiana University, Tech. Rep. DOTHS-034-3-535-77, 1977.

[3] M. A. Brewer, G. Pesti, and W. H. S. IV, "Improving compliance with work zone speed limits: Effectiveness of selected devices ", Transportation Research Record, vol. 1948, pp. 6776, 2006.

[4] "Fatalities in motor vehicle traffic crashes by state and construction/maintenance zone ", http://wzsafety.tamu.edu/crash data/workzone fatalities/2006.

[5] A. Arora et al. Exscal: Elements of an extreme scale wireless sensor network. In *RTCSA '05*, pages 102–108, 2005. IEEE.

[6] A. Awad, T. Frunzke, and F. Dressler. Adaptive distance estimation and localization in wsn using rssi measures. In *DSD '07*, pages 471–478, 2007. IEEE CS

[7] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *SenSys 2008*, November 2008.

[8] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Par-

lange. Sensorscope: Out-of-the-box environmental monitoring. In *IPSN '08*, pages 332–343, 2008. IEEE Computer Society.

[9] California Center for Innovative Transportation. Traffic surveillance. http://www.calccit.org/itsdecision/serv_and_tech/Traffic_Surveillance/surveillance_overview.html.

[10] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar. Brimon: a sensor network system for railway bridge monitoring. In *MobiSys '08*, pages 2–14, 2008. ACM.

[11] S.-Y. Cheung and P. Varaiya. Traffic surveillance by wireless sensor networks: Final report. Tech Rep UCB-ITS-PRR-2007-4, California PATH Program, Inst of Transportation Studies, UC, Berkeley, 2007.

[12] Devantech. SRF02 ultrasonic range finder. http://www.robot-electronics.co.uk/htm/srf02techI2C.htm, 2008.

[13] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[14] S. F. Duffy. CSU UTC. http://www.csuohio.edu/utc, 2007.

[15] Federal Highway Administration. *Manual on Uniform Traffic Control Devices*. U.S. Department of Transportation, Washington D.C., 2003 edition with revisions 1 and 2 edition, December 2007.

[16] Federal Highway Administration. Benefits of using intelligent transportation systems in work zones. Technical report, US Department of Transportation, April 2008.

[17] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems. In *PLDI '03*, pages 1–11, 2003. ACM.

[18] Google. Google maps. http://maps.google.com, 2008.

[19] A. Haoui, R. Kavaler, and P. Varaiya. Wireless magnetic sensors for traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 16(3):294–306, June 2008.

[20] T. Herman and C. Zhang. Stabilizing clock synchronization for wireless sensor networks. In *8th SSS*, pages 335–349, 2006.

[21] Dheeraj R. Bheemidi and Nigamanth Sridhar. A Wrapper-Based Approach to Sustained Time Synchronization in Wireless Sensor Networks. In Proceedings of the 17th International Conference on Computer Communications and Network s. August 3–7, 2008. St. Thomas US Virgin Islands. Aug 4–7 2008.

[22] J. Hill et al. System architecture directions for networked sensors. In *ASPLOS-IX*, pages 93–104, 2000. ACM.

[23] M. M. Holland, R. G. Aures, and W. B. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In *SECON '06*, 2006.

[24] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *SenSys '06*, pages 125–138, 2006. ACM.

[25] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: a reliable bulk transport protocol for multihop wireless networks. In *SenSys '07*, pages 351–365, 2007.

[26] A. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. Tech report, MIT, May 2000.

[27] P. Levis. Tinyos tep 111: `message_t`. http://www.tinyos.net/tinyos-2.1.0/doc/html/tep111.html, 2008.

[28] David Moss et.al. Tinyos tep 126: `CC2420 Radio Stack`. http://www.tinyos.net/tinyos-2.1.0/doc/html/tep126.html, 2008.

[29] Microsoft. Microsoft live maps. http://maps.live.com, 2008.

[30] P. Mohan, V. Padmanabhan, and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys 2008*, November 2008.

[31] T. Nadeem, S. Dashtinezhad, and C. Liao. Trafficview: A scalable traffic monitoring system. In *2004 IEEE ICMDM*, pages 13–26, 2004.

[32] S. Nath, J. Liu, J. Miller, F. Zhao, and A. Santanche. Sensormap: a web site for sensors world-wide. In *SenSys '06*, pages 373–374, New York, NY, USA, 2006. ACM.

[33] Ohio Department of Transportation. Buckeye traffic information. http://www.buckeye-traffic.org, 2008.

[34] J. Polastre. Telos charger board. http://www.tinyos.net/hardware/telos/sensorboards/UCB_TSB+TCB_schematic-2005-07-31.pdf, 2005.

[35] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05*, page 48, 2005. IEEE Press.

[36] G. Santel. Traffic flow and accident occurrence in construction zones on freeways. In *6th Swiss Transport Research Conf*, March 2006.

[37] Sharp. Sharp GP2Y0A700K0F IR sensor data sheet. http://www.acroname.com/robotics/parts/R302-GP2Y0A700K0F.pdf.

[38] MAXIM. MAX756 Step up converter data sheet. http://pdfserv.maxim-ic.com/en/ds/MAX756-MAX757.pdf.

[39] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley. Tracking multiple targets using binary proximity sensors. In *IPSN '07*, pages 529–538, 2007. ACM.

[40] K. Srinivasan and P. Levis. Rssi is under-appreciated. In *EmNets*, 2006.

[41] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and A. Ledeczi. Shooter localization and weapon classification with soldier-wearable networked sensors. In *MobiSys '07*, pages 113–126, New York, NY, USA, 2007. ACM.

[42] M. Welsh, G. Werner-Allen, K. Lorincz, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Sensor networks for high-resolution monitoring of volcanic activity. In *SOSP '05*, pages 1–13, 2005. ACM.

[43] K. Whitehouse, F. Zhao, and J. Liu. Semantic streams: A framework for composable semantic interpretation of sensor data. In *EWSN*, pages 5–20, 2006.

[44] J. Yoon, B. Noble, and M. Liu. Surface street traffic estimation. In *MobiSys '07*, pages 220–232, New York, NY, USA, 2007. ACM.

[45] Sing-Yiu Cheung and Pravin Varaiya. Traffic Surveillance by Wireless Sensor Networks: Final Report. UCB-ITS-PRR-2007-4 California PATH Research Report.

[46] Donal McErlean and Shrikanth Naryanan. Distributed Detection and tracking in sensor networks. In 36th Asilomar Conf.Signals,Systems and Computers, 2002.

[47] Datasheet, Crosbow Technology, Mica2. http://www.xbow.com/

[48] Sun SPOT. Sun SPOT information. http://www.sunspotworld.com/

[49] Imote2. Datasheet, Crosbow Technology, IMote2. http://www.xbow.com/

[50] CC2420 Datasheet. 2.4 GHz IEEE 802.15.4 ZigBee-ready RF Transceiver. http://focus.ti.com/lit/ds/symlink/cc2420.pdf/

[51] Chipcon, CC1000 Single Chip Very Low Power RF Transceiver 2002. http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf/

[52] TDA5250 D2 ASK/FSK 868MHz Wireless Transceiver http://www.infineon.com/