Fighting Chaotic Road Congestion
Second PhD Annual Progress Report

Rijurekha Sen
under the guidance of
Prof. Bhaskaran Raman

January 17, 2011

Abstract

Road congestion is a common problem worldwide. In developed countries, automated congestion detection techniques are used in road travel assisting applications. But these techniques are mostly inapplicable in developing regions due to high cost and their assumptions of orderly traffic. We present two techniques for effective congestion detection in chaotic traffic of developing regions.

Our first system is RoadSoundSense, an acoustic sensing based road congestion monitoring technique. We detail the design of a prototype which, deployed by the roadside, processes road noise and sends various metrics to a remote server. Data from deployment of this prototype on six Mumbai roads, validated against manually observed ground truth, shows feasibility of per minute congestion monitoring. K-means clustering gives 90% accuracy on average, in training our system to work on a new road. Deployment data from one road for six days shows the temporal variation in traffic state for that road.

We also present WirelessAcrossRoad, which exploits the variation in wireless link characteristics when line of sight conditions between a wireless sender and receiver vary. The system comprises of a wireless sender-receiver pair across a road. The sender continuously sends packets. The receiver measures metrics like signal strength, link quality and packet reception. These metrics show a marked change in values depending on traffic-state on the road in between. From about 15 hours of data on two different Mumbai roads, we show that we can classify traffic states as free-flowing and congested using a decision tree based classifier with 97% accuracy.

Chapter 1

Introduction

Road traffic problems - The average number of vehicles on Indian roads is growing at an enormous rate — 10.16% annually since the last five years [1]. Mumbai, a metropolitan city, has over 590 vehicles per km of road. Bangalore, another metropolitan city, has about 5 million vehicles plying on a road network of barely 3000 kms [2, 3]. This is leading to increasing levels of road congestion, longer and unpredictable travel times and wastage of time and fuel for commuters. Growth in infrastructure has been slow due to various reasons such as high cost, lack of space, bureaucracy, etc.

We define traffic on a road to be *free-flowing*, if each vehicle moves at the desired speed of its driver, bounded by the speed limit of the road. Any situation other than this, where drivers have to slow down vehicles or stop and go, because of presence of several other vehicles on road, is considered *congested*. Automated congestion detection can help in traffic management by making traffic signal timings more efficient. Secondly, if some roads or junctions show regular trends of becoming congested, this knowledge can be used to plan new infrastructure such as flyovers and freeways. Thirdly, this can be used to design interesting mobile applications for on road commuters. Which congested routes should be avoided? What is the travel time along a congested route? Applications that can answer such questions will certainly make traveling on roads less cumbersome.

Issues with existing solutions - Many congestion detection techniques [4, 5, 6, 7, 8, 9] are already being used in developed countries. But unlike traffic in developed countries, traffic on Indian city-roads is characterized by high variability in size and speed of vehicles [10, 11]. The same road is shared by buses, trucks, cars, vans, auto rickshaws, motor-bikes, bicycles, and pedestrians. Traffic is often chaotic, with no semblance of a lane-system common in developed countries [12]. Thus, as we discuss in Section 2, the various congestion detection techniques used in developed countries will not be directly applicable in an Indian context.

RoadSoundSense: Acoustic sensing based congestion monitoring - Unlike traffic in developed countries, chaotic traffic in developing regions is very noisy. One of the characteristic sounds comprising this noise is vehicular honks, which drivers use excessively to alert other drivers and pedestrians. We have made a list of about 50 video clips [13], shot on Indian roads, from which the noisiness of chaotic traffic becomes apparent. In *RoadSoundSense*, we seek to exploit this distinctive feature of ”excessive noise” to do congestion detection on chaotic roads.

In our prior work [14], vehicle honks recorded in roadside recorders were used to estimate vehicle speeds using differential Doppler shift. Metrics like vehicle speed distribution and amount of vehicle honks gave 70% accuracy in classifying traffic state into congested and free-flow using threshold based classification. But can this technique, involving computation intensive acoustic signal processing, be implemented on an embedded sensor platform, to be used for on-road sensing? Can the sensing and processing be done in near real time? Will the cost be low enough? These are some *implementability issues* of *RoadSoundSense*. Will the system be able to detect congestion on a wide variety of roads? Will the traffic classification model vary from road to road? In that case, what will be the training overhead of our system on a new road? Can we do without training using unsupervized learning? These are the *usability issues* of *RoadSoundSense*.

In this work, we seek to answer the above questions. We present the detailed design of an acoustic sensing hardware prototype which has been deployed by the side of the road. This unit samples and processes road noise to compute various metrics like amount of vehicular honks and vehicle speed distribution and sends the metrics to a remote server every alternate minute. Data from deployment of this prototype in six different Mumbai roads, validated against manually observed ground truth, shows feasibility of per minute congestion monitoring from the remote server. K-means clustering gives on average 90% accuracy to group unlabeled data on a new road into two clusters of congested and free-flow. Deployment data from one road for six days shows the temporal variation in traffic state for that road. Our prototype has a moderate cost of $160 and is easy to install and maintain on road-side lamp-posts.

WirelessAcrossRoad - RF sensing based congestion detection - In a different field of work, in the area of wireless networks, prior literature [15, 16, 17] shows that wireless link behavior suffers in absence of clear line of sight between the sender and receiver. In this work, we exploit this prior knowledge and design a new congestion detection technique that can handle chaotic traffic. Our technique comprises of a wireless sender-receiver pair across a road. The sender continuously sends packets. The receiver measures metrics like signal strength, link quality and packet reception. We show that these metrics show strong correlation with free-flowing or congested traffic states of the road. Our technique gives 97% traffic classification accuracy on 15 hours of data collected from two different roads in Mumbai. The initial cost analysis shows that our system will need about $200 per installation and will be easy to deploy on roadside lamp-posts.

Chapter 2

Related Work

Road congestion detection needs on road sensing and there are several existing congestion detection techniques based on a variety of sensors. Since our contribution so far, has been in designing, implementing and testing two new on-road sensing techniques, we have to compare them to the existing techniques. The comparison has to be based on effectiveness in detecting congestion in chaotic traffic, at a low cost, causing minimal disturbance to traffic while installation and maintenance. Proliferation in developing countries and incentive and privacy issues of participatory sensing are other factors to consider. We present such a comparison in the Table 2.1.

	Dual loop detector	Magnetic sensing	Image sensing	GPS in public transport	Smartphones	Acoustic sensing	RF sensing
Handles non-laned traffic	no	no	yes	yes	yes	yes	yes
Cost (infrastructure/computation)	high	low	high	low	low	low	low
Disruption while installation/ maintenance	yes	yes	no	no	no	no	no
Needs commuters' participation	no	no	no	no	yes	yes	no
Commuters incur cost	no	no	no	no	yes	no	no
Commuters have privacy issues	no	no	no	no	yes	no	no
Proliferation in developing countries	low	low	low	low	moderate	–	–

Table 2.1. Comparison table of our techniques with state of the art in road sensing

Fixed sensor based techniques - In these techniques, the sensors that gather various road related information are statically placed on or by the side of the road. Examples are dual loop detectors [4], magnetic sensors [6] and image sensors [5]. These techniques can be prohibitive in terms of infrastructure and maintenance costs. As given in [18], the initial installation cost of a vehicle loop detector is approximately $26,100. Secondly, the inherent assumption of lane-based orderly traffic makes these techniques inapplicable for chaotic road conditions. Thirdly, the assumption of low variability in vehicle speeds also does not hold in developing regions where heavy slow moving trucks and high speed motorbikes ply on the same road at the same time. Finally loop and magnetic detectors need to be placed under the road. If roads need to be dug up to install and maintain the infrastructure, that will adversely affect traffic, as alternate routes are often times unavailable. Moreover, road lifetime in India is less than ideal. Each time the road is relaid, the sensors will have to be reinstalled. With regards to image processing, high resolution Point-Tilt-Zoom (PTZ) cameras are expensive and the image processing complexity to handle disorderly traffic is high.

Probe vehicle based techniques - In these techniques, the sensors are mobile and placed in a subset of vehicles that ply the road. Examples are GPS-enabled probe-vehicles [7, 8, 9] and multiple sensor enabled smartphones in vehicles [12]. In India, proliferation of GPS receivers in vehicles is quite low. Only a small number of taxi fleet companies and state transport companies have GPS units installed in their vehicles in a few metropolitan cities. Smartphone penetration in India is also quite low [19], though mobile phone penetration is extremely high. Most people have low end phones and are unable to take part in participatory sensing. Even for those who have smartphones, it is difficult to think of an incentive model to attract them to take part in GPS sensing as it involves sensing as well as communication costs.

Our two techniques: acoustic and RF sensing - Both our sensing techniques are suitable in all ways for congestion monitoring on chaotic roads. Only acoustic sensing has the drawback of depending on noise from vehicles. Quality control of vehicle engines and tyres and also of road pavement are not strictly observed due to issues of funds and bureaucracy and lack of public awareness increasing road noise in developing regions. Also, the non-lane based disorderly traffic causes excessive use of vehicle brakes and honks. [20, 21, 22, 23] study the high amount of noise pollution in the four Indian cities of Asansol, Jaipur, Kolkata and Varanasi respectively, and such pollution is common in other Indian cities too. But still the dependency on vehicle emitted noise makes the acoustic sensing based technique less ubiquitous. Secondly, it deviates from the public sentiment against noise pollution. *Though we are using acoustic sensing, our aim is not to promote noise pollution.* We seek to use an already existing negative feature for the positive purpose of road congestion monitoring. And as our extensive experimental results on six Mumbai roads show, the technique is highly effective in congestion monitoring on most busy urban roads.

Chapter 3

RoadSoundSense: Acoustic Sensing Based Congestion Monitoring

3.1 My Prior Work: ”Horn-Ok-Please”

In [14], we developed a technique to estimate vehicle speed using differential Doppler shift of vehicle honks recorded at roadside recorders. The system architecture is given in Fig. 3.1. Suppose that a sound source moves with speed v_s, and the receiver (observer) is stationary. Denote the emitted audio frequency as f_0 and speed of sound as v. When the source is moving away from the receiver, the frequency observed at the receiver is given by,

$$f_1 = \frac{v}{(v + v_s)} f_0 \tag{3.1}$$

And when the source is moving towards the receiver, the frequency observed at the receiver is given by,

$$f_2 = \frac{v}{(v - v_s)} f_0 \tag{3.2}$$

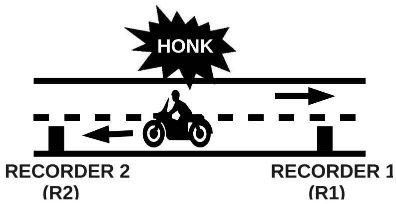


Figure 3.1. Original System Architecture

If f_0 is known, v_s can be estimated easily from Eqn. 3.1 or Eqn. 3.2, and one sensor would suffice. But it is not easy to guess f_0, as different honks have different base frequencies. We thus used a two-sensor architecture: Fig. 3.1 depicts a deployment of two recorders by the side of a two-way road. When a moving vehicle blows honk in between the two receivers, it is approaching one receiver and receding from the other. Substituting the value of f_0 from Eqn. 3.1 in Eqn. 3.2, we get following equation,

$$v_s = \frac{(f_2 - f_1)}{(f_2 + f_1)} v \tag{3.3}$$

In [14], we developed algorithms for **(1) Honk detection:** The two recorders record and *detect* honk from the noisy road recording independently. **(2) Honk matching:** We then have to *match* same honks between the two recorders, so that we apply Eqn. 3.3 for the same honk. **(3) Frequency extraction:** We have to *extract* f_1 and f_2 and apply Eqn. 3.3 to get the speed estimate. Using over 18 hours of recordings from two roads in Mumbai, we showed that there are enough honk samples for our method to be useful and that our speed estimation technique is effective in real conditions. Further, we used our data to characterize traffic state as free-flowing versus congested using a variety of metrics: the vehicle speed distribution, the amount of honks and noise level. We used the voice recording application N79 phones for roadside recordings. Two persons had to be present for hours on the road to manually record road noise. Back in lab, the hour long recordings would be analyzed to get the information described above.

Contributions of this report:

(1) We design and develop an acoustic sensing hardware prototype, comprising of Recorder1 (R1) and Recorder2 (R2), and show that computation intensive acoustic signal processing and reliable data communication to remote server in near real time are feasible on resource constrained embedded platform. The enhanced system architecture is shown in Fig. 3.2, where each prototype implements the architecture given in Fig. 3.1.

(2) Data from deployment of this prototype in six different Mumbai roads gives valuable insight into choosing appropriate acoustic metrics for congestion detection on particular roads.

(3) We address the problem of unsupervised learning of the traffic classification model by our unit, on a new road, using K-means clustering.

(4) Deployment data from one road for six days shows the temporal variation in traffic state for that road.

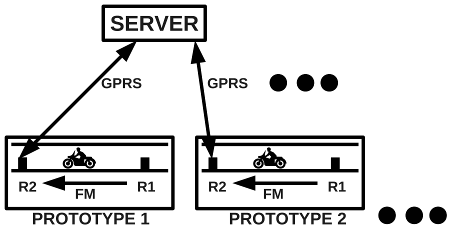


Figure 3.2. Enhanced System Architecture

3.2 Prototype Design

Our two sensor based congestion detection technique, that this prototype has to implement, has three functional requirements - (1) **Sensing** - sampling road noise, (2) **Processing** - filtering of sensed noise, detecting honks, matching honks between R1 and R2 after time synchronization and detecting speeds and (3) **Remote communication** - sending various metric values like amount of honks and speeds to remote server.

3.2.1 Design choices

The design choices available to us for R1 and R2 are as follows.

Both R1 and R2 having sensing and remote communication capabilities. Sampled raw audio signal will be sent to a central server which will do the processing. The primary disadvantage here is that, the amount of data to be communicated to the central server will be huge, though central server would need only the honk related information. This will unnecessarily increase communication cost and delay.

Both R1 and R2 having sensing, computation and remote communication capabilities. Only computed metric values will be sent to the server. The two units will also need local communication capabilities to communicate between each other, as honk matching and speed calculation cannot be done individually. This seems plausible but has scope of further optimization as specified in the next design choice.

R1 having only sensing and R2 having sensing, computation and remote communication capabilities. The units will have local communication capabilities for R1 to send sampled analog signal to R2 over a small inter-sensor distance. R2 will do processing and remote communication. This choice seems good, provided we can handle the various technical challenges discussed next.

First, the quality of audio signal should not be affected during communication from R1 to R2. Secondly, absence of computation capabilities on R1 would prevent time stamping of the audio signal. But R2 should know which sensing event happened when for matching honks and sending time-stamped metrics to server. So propagation delay of audio signal from R1 to R2 has to be negligible. Thirdly, R2 should have enough capabilities to perform computation intensive operations like FFT on two audio signals. We now show how we choose our hardware components to meet these challenges.

3.2.2 Hardware

Recorder 1 (R1) - To meet the first two technical challenges, we take idea from any commercially available wireless microphone, that has an FM transmitter to transmit voice signal to the amplifier, which has an FM receiver. We seek to use FM to transfer analog signal from R1 to R2. Using commercially available WR-601 FM transmitter-receiver pair [24], we achieve good quality analog signal transmission, i.e. the frequency spectrum of original signal and signal transmitted over FM does not vary.

We experimentally test propagation delay using two recorders - an N79 phone and a laptop with an FM receiver. A square wave pattern is played from a speaker. The N79 phone and an FM transmitter with microphone are placed very near to the speaker. The laptop with FM receiver is first held close to the speaker and then gradually moved to 25m, the square wave pattern being played every 1m. Synchronizing the first recorded wave pattern in the laptop and the N79, to remove any time offset in starting the two recorders, synchronizes all 25 recorded patterns, with 3μs offset only in one case. The offset is 0μs in all other cases. This shows that analog signal propagation delay over FM is negligible.

But the range obtained using FM with line of sight between the transmitter and the receiver is around 25m, and not 30m, as used in [14]. Smaller range means less distance between R1 and R2, reducing number of vehicles honking between them. But since the advantages of using FM in terms of hardware complexity reduction, far exceed this small drawback, we decide to use it.

Another component required at R1 is the microphone, for which we use CTP-10DX miniature omnidirectinal 15 m range electret condenser microphone from Ahuja [25]. Fig. 3.3 shows our prototype hardware for R1.

Recorder 2 (R2) - Acoustic signal processing motivates the use of a Digital Signal Processor (DSP) instead of an ordinary microcontroller. We use an evaluation module from TI, C5505 ezdsp [26], which has a small form factor (3.15 x 1.5 inches), C5505 DSP chip with 100 MHz clockrate at active power less than 0.15 mW/MHz, AIC3204 codec chip, a line-in socket and a 60 pin expansion connector including SPI and UART pinouts. It also has an FFT Hardware Accelerator, a tightly coupled coprocessor, that we use extensively in our audio signal processing. We use off-the-shelf GPRS modem having SIM300 GSM/GPRS module from SIMCOM [27], with serial connectivity options. We include a flash memory in our unit, in case computed data needs to be stored before sending. For this we use 128 Mbit SPI flash from Spansion [28]. The same microphone, as used in R1, is used in R2 too. We design a PCB to interface the different components which also has the powering circuitry described next. Audio connections for R2 are shown in Fig. 3.4 and non-audio connections in Fig. 3.5. Fig. 3.6 shows the overall hardware block diagram comprising of R1 and R2.

Power supply - The GPRS modem oprates at 12V with 2A peak current. The DSP module operates at 5V with peak current in mA. The flash has an operating voltage of 2.7-3.6 V with 26 mA peak current. We use a pack of 6 Lithium-ion batteries, with 12V output

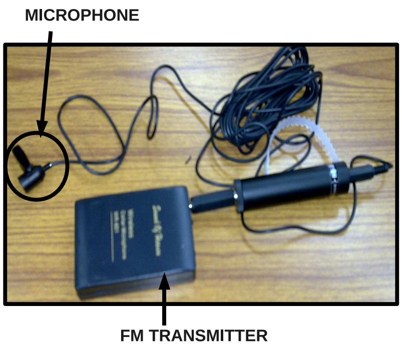


Figure 3.3. R1

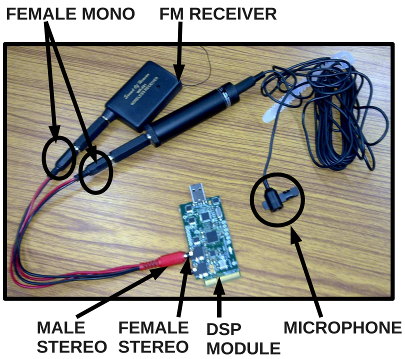


Figure 3.4. R2: Audio Connections

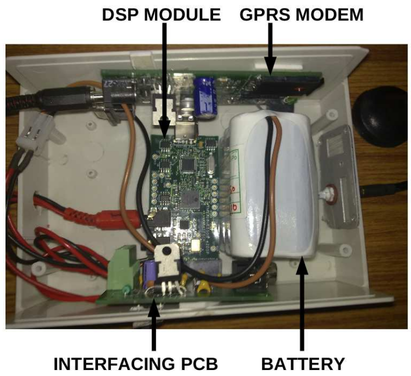


Figure 3.5. R2: Non-Audio Connections

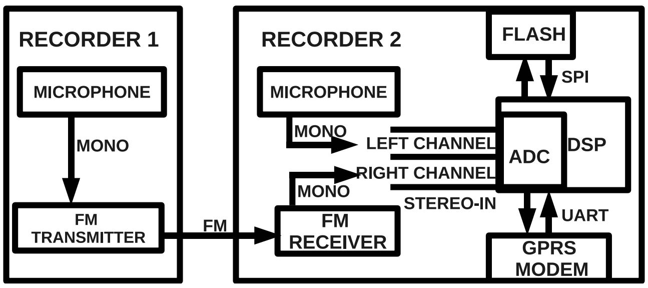


Figure 3.6. Block Diagram of Hardware

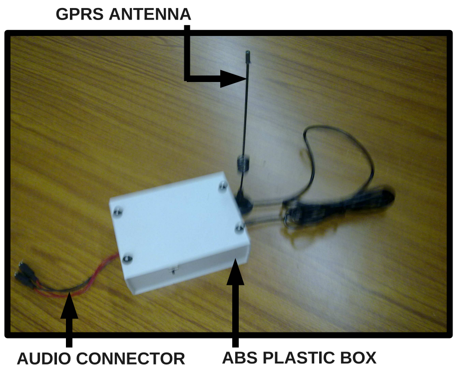


Figure 3.7. R2 Packaging

voltage, 3A peak current and 4AH capacity. The interfacing PCB takes input from battery, directly powers up the GPRS modem and feeds the DSP module through a AP1512AT5 DC-DC converter [29] to step down from 12V to 5V. The flash is powered up from VCC pin of DSP module supplying 3.3V.

Enclosure - As shown in Fig. 3.7, we use a 15.5cm x 12cm x 6.5cm waterproof ABS plastic enclosure [30] for R2. The audio connectors to plug in the microphone and the FM receiver and the GPRS antenna remain outside the enclosure.

Cost - Table 3.1 gives the cost breakup for the different components. All the components are off-the-shelf modules, which has somewhat increased the cost. If we make our own custom board, with only the required ICs and connectivity, we believe that the cost can be kept under $100. The GPRS communication cost is Re 0.1 per 10 KB and we send about 50 bytes of data per minute. Hence this cost is negligible.

Item	Unit Price($)	Quantity	Cost($)
DSP module	50	1	50
GPRS modem	50	1	50
FM tx-rx	15	1	15
Microphone	5	2	10
Interfacing PCB	5	1	5
Battery	20	1	20
Enclosure	5	1	5
Flash	3	1	3
Connectors	0.4	5	2
Total			160

Table 3.1. Prototype Cost Breakup

3.2.3 Software

The DSP module comes with a CCStudio IDE [31] for programming and debugging. The boot image of user program is loaded to 64 KB EEPROM. In our program, 16-bit stereo (composed of two mono signals as shown in Fig. 3.6) samples are captured by the AIC3204 codec at 16 KHz sampling frequency and copied to the DSP memory using DMA. We use ping-pong buffering to avoid missing samples during processing. Processing involves filtering of the noisy signal, detecting honks in each individual mono channel, matching honks between two channels and extracting $f1$ and $f2$ to compute speed. The details of detection, matching and frequency extraction algorithms can be found in [14].

The sequence of events that happen after powering up our unit is as follows. The flash memory is erased. The DSP sends a request for current time-stamp to the remote server over GPRS and initializes the RTC based on the received time-stamp. Then the following loop starts. Audio is sampled, filtered and honks are detected for time t_{sample}; honks are matched, speeds are calculated and metrics are stored in flash for time $t_{process}$. This continues for time $t_{workLocal}$. Next in time $t_{sendRemote}$, metrics are read from the flash and sent to the remote server over GPRS. This loop of sampling, processing, storing for time $t_{workLocal}$ and sending for time $t_{sendRemote}$ continues indefinitely.

t_{sample}, $t_{process}$, $t_{workLocal}$ and $t_{sendRemote}$ are parameters, whose values are chosen either by resource constraints or by repeated experiments to detect worst case processing delay. For example, t_{sample} is decided to be 4 secs because of resource constraint. In honk detection, we use 128 point FFT, for which 128 audio samples are needed. Each audio sample is 2 bytes. In 1 sec, at 16KHz sampling frequency, there are 125 windows of 128 samples. Each window has a 2 byte time-stamp (10 bits for millisecs and 6 bits for secs). During detection, we store the time windows having honk characteristics, with time-stamp, to be used later for honk matching and frequency extraction. In the worst case, each of the 125 windows can have honk characteristics, for which we have to store (((128x2)+2)x125) bytes/sec = 32KB/sec. For 2 channels, we need to store 64KB/sec. Given a 320KB RAM, after setting aside storage for code, stack and temporary variables, we can use about 256KB to store honk windows. This limits t_{sample} to (256KB)/(64KB/sec) = 4 secs. $t_{process}$ is set at 5μs, after repeated experiments to ascertain flash writing time. $t_{workLocal}$ is set at 1 min, because we want updates at the server at least every alternate minute. $t_{sendRemote}$ is set at 50 secs, after repeated experiments to ascertain combined time for reading flash, setting up new TCP socket connection over GPRS in case there is no existing connection and sending data.

3.3 Prototype Deployment

Our hardware prototype is moderate in cost and is able to sample and process sound and send metrics to server every alternate minute. This solves the *implementability issues* of our technique. We next seek to answer the questions pertaining to the *usability issues*. Will acoustic metrics portray traffic state on a wide variety of roads? Should choice of metrics be road specific? In that case, will standard machine learning tools like Support Vector Machine (SVM) be able to automate the metric choice for a particular road? Labeling data instances with classes for training, generally involves human judgment, hence it is costly and time consuming. So, will unsupervised learning techniques (which do not require training data) be able to find patterns in the data for our system?. We deploy our prototype on different roads in Mumbai to seek answers to these questions.

3.3.1 Deployment locations

The server is kept within IIT Bombay campus in Powai. The hardware prototype units are deployed at six locations in Mumbai. The locations are listed in Table. 3.2 and their positions on Googlemap of Mumbai are shown in Fig. 3.8. The locations are near important road junctions or railway stations, where congestion occurs daily during the peak hours. The roads have variable width and traffic type. The locations are chosen within a distance of 5-6 Km from IIT to reduce trip time from our lab to each location.

The positioning of the prototype on a road, in relation to the location of the traffic signal along that road, matters significantly. Vehicles are expected to stop at the signal, so standing traffic very near to signal is not considered congested. On the other hand, traffic queue length on that road never exceeds a certain limit even in the worst case. Thus the road stretch very far from signal never gets congested. So the prototype should be at an intermediate position where congestion monitoring makes sense. Also, it does not make sense to put the prototype after traffic signal, as there is mostly no congestion there. In our deployments, we use a distance of 150-200m before a traffic signal, after manually observing the queue length on that road.

No.	Location	Road bi-directional	Road width (each way)	Vehicle type
1	Bhandup	Yes	10m	All+
2	Vikhroli	Yes	10m	All+
3	Gandhinagar	Yes*	25m	All+
4	Chandivali	Yes	15m	All+
5	Ghatkopar	Yes	10m	All+
6	Powai (Hiranandani)	Yes	8m	Light

Table 3.2. Deployment Location Details

* Road width prevents sensing of noise from opposite direction.

+ Heavy (trucks, buses) and light (cars, motorbikes, autorickshaws)

To avoid the hassles of getting permission from city authorities to put up the units on lamp-posts, we deploy them at road-side shops. We use locks with steel chains through clamps, to prevent stealing of the deployed units.

3.3.2 Results

We program our prototype to send the following timestamped values computed over 1 minute to the remote server in the next minute - (1) number of honks in R1 (numhonks1), (2) duration of honks in R1 (duration1), (3) number of honks in R2 (numhonks2), (4) duration of honks in R2 (duration2) and (5) all vehicle speed samples. The server uses the first four as *honk-based metrics*. From the last, it computes 70^{th} percentile speed (70speed) and percentile speeds less than 10 Kmph (10perc), the *speed based metrics*.

From 22^{nd} to 27^{th} Nov, 2010, we remain for 2-3 hours on road, manually observing the traffic state, one day at each deployment location. Two videos of traffic for each location, one showing free-flow and the other showing congestion, can be found at [13]. Fig. 3.9 - Fig. 3.20 show plots of first three deployment locations. The captions signify {Location, time}. Each vertical bar represents 1 minute and we color code the bars according to our manual observation. The horizontal black lines are provided in each plot to aid the height comparison among bars. As we can see, number and duration of honks are higher in congestion than in free-flow. In congestion, 70^{th} percentile speeds are low, while percentile speeds less than 10 Kmph are high.

Road specific choice of metrics

Though all four metrics somewhat correspond to manually observed traffic state, there are some metrics which bring out the traffic state at particular locations, more accurately than other metrics. Here we seek to intuitively understand the reasons behind this, as it will help

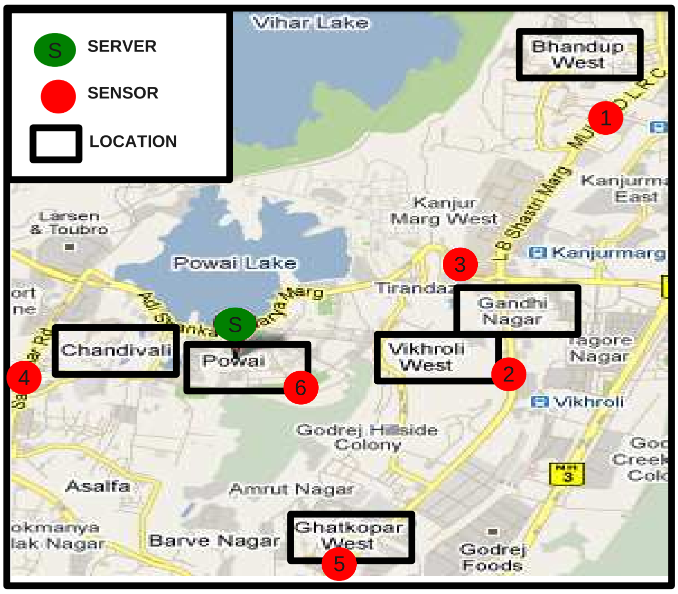


Figure 3.8. Sensor Deployment Locations in Googlemap

Bhandup (Accuracy 93.2%)	Vikhroli(Accuracy 98.3%)	Gandhinagar (Accuracy 100%)
$+1.38 * numhonks1$	$+0.71 * numhonks1$	$+1.45 * numhonks1$
$+0.38 * duration1$	$-0.21 * duration1$	$+0.97 * duration1$
$-0.17 * numhonks2$	$+0.15 * numhonks2$	$+1.59 * numhonks2$
$+1.19 * duration2$	$-0.57 * duration2$	$+0.91 * duration2$
$-2.94 * 70speed$	$-2.71 * 70speed$	$-0.58 * 70speed$
$+2.25 * 10perc$	$+2.71 * 10perc$	$+0.49 * 10perc$
-1.73	-0.94	-1.98

Table 3.3. Attribute Weights using Binary SMO with Linear Kernel

us in selecting good attributes to build traffic classification model for each location.

(1) Fig. 3.9, Fig. 3.12, Fig. 3.15 and Fig. 3.18 show good difference between two traffic-states at Bhandup based on all four metrics. This is found in the other three deployment locations of Chandivali, Ghatkopar and Powai (Hiranandani) also. But these roads are bidirectional and not very wide, so honks from opposite direction are recorded too. The speed based metrics are direction sensitive, as Equation 3.3 gives signed speeds based on direction. Hence we can filter out speed values in opposite direction. But we cannot do this for honk based metrics, which might give inflated values. Thus though all four metrics are good for these locations, speed based metrics should be given higher weightage than honk based metrics in any road state classification.

(2) As seen in Fig. 3.10 and Fig. 3.13, the honk based metrics show no difference between congested and free-flowing traffic states at Vikhroli. This is because, just in front of the shop where our unit is deployed, there is a small cut in the divider between the bi-directional road. The pedestrians use this for road crossing. Thus, even in free-flowing traffic, many vehicles blow honk for alerting pedestrians and the road being quite busy, number and duration of honks in free-flow is high. But speed based metrics bring out the difference accurately here, as ample honking gives good number of matched honks and speeds, whose values vary widely between free-flow and congested. Thus only speed-based metrics are good for this location.

(3) As seen in Fig. 3.11 and Fig 3.14, the honk based metrics like number and duration of honks show clear difference between two traffic states at Gandhinagar. The width of the road being 25m for this location, number of vehicles increases almost five-fold in congestion from free-flow. Hence number and duration of honks go up drastically. The speed based metrics are more complex to calculate than the honk based metrics as it involves matching honks between R1 and R2 and computing 1024 point FFT for frequency extraction. So using only honk based metrics will be good for this location.

Whether standard classification models like SVM, built using training data for a particular road, incorporates this road-specific weightage of attributes, is investigated next.

Classification models

We build a training set for each road, having 90 instances for Bhandup, 58 instances for Vikhroli and 60 instances for each of the other four locations. Each instance of the training set has 6 attributes, four honk-based and two speed-based metrics, and 1 class label, based on manual observation. We use WEKA, a widely used open source package for machine learning tools [32]. We input each training set to WEKA and get a binary Sequential Minimal Optimization (SMO) SVM model with linear kernels as output. The models for Bhandup, Vikhroli and Gandhinagar are given in Table 3.3. We highlight some cells to show that the weights assigned to the attributes are in accordance with our discussion in the previous section. For Bhandup and Vikhroli, the speed based metrics are given more weightage than the honk based. For Gandhinagar, the honk based metrics are given more weightage. Thus standard machine learning tools, can incorporate the road specific metric choice in the traffic classification model for that road, with some amount of training data.

The accuracy values for 10-fold cross validation are also given in the table. This accuracy is minimum (92.7%) for Hiranandani (not shown in the table).

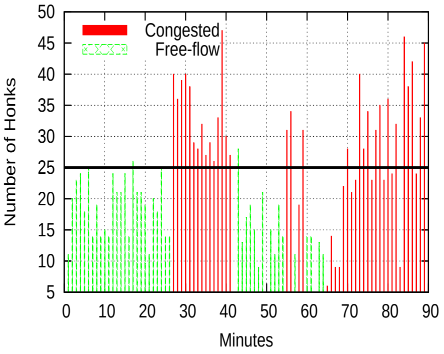


Figure 3.9. Bhandup, 5:30-8:30pm

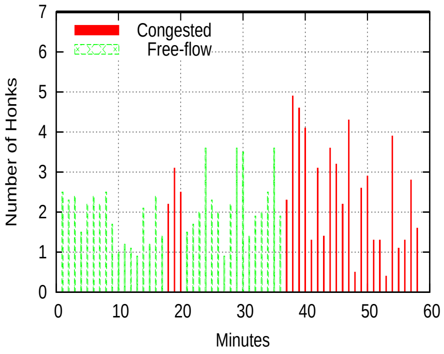


Figure 3.10. Vikhroli, 5:45-7:40pm

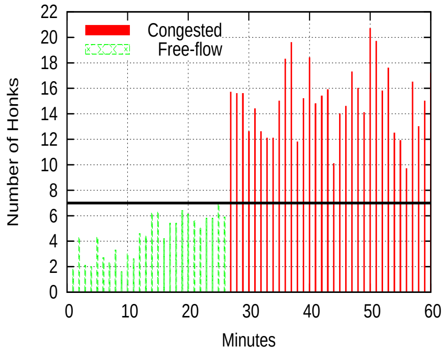


Figure 3.11. Gandhinagar, 6-8pm

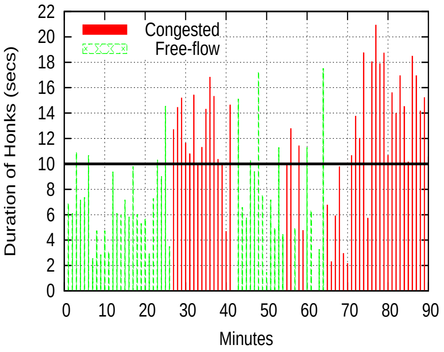


Figure 3.12. Bhandup, 5:30-8:30pm

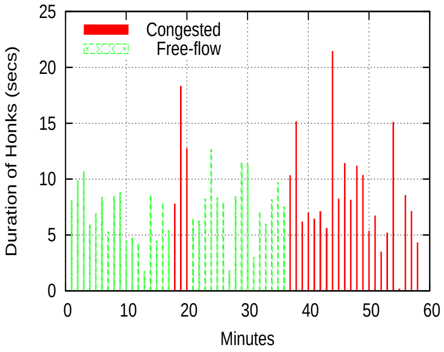


Figure 3.13. Vikhroli, 5:45-7:40pm

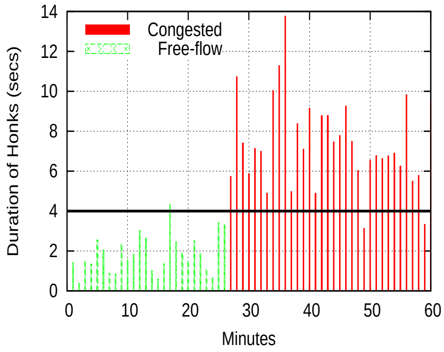


Figure 3.14. Gandhinagar, 6-8pm

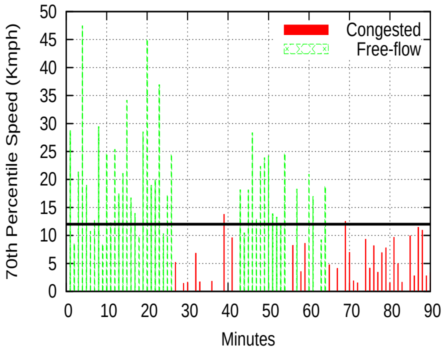


Figure 3.15. Bhandup, 5:30-8:30pm

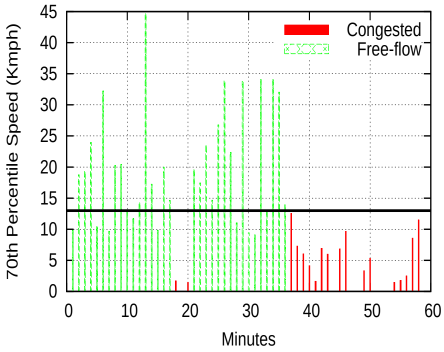


Figure 3.16. Vikhroli, 5:45-7:40pm

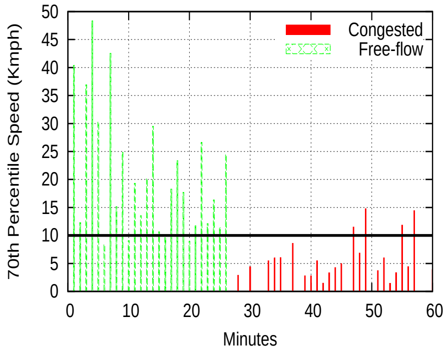


Figure 3.17. Gandhinagar, 6-8pm

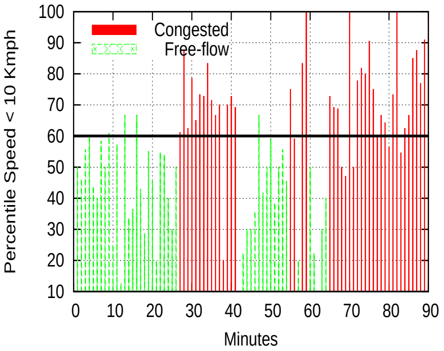


Figure 3.18. (1), 22/11, 5:30-8:30pm

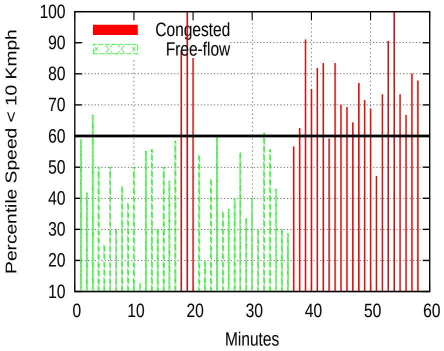


Figure 3.19. (2), 23/11, 5:45-7:40pm

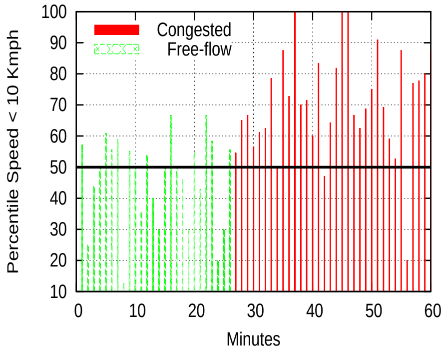


Figure 3.20. (3), 24/11, 6:00-8:00pm

Self-learning on new road

Next we seek to address the problem of absence of any training data, if we deploy our unit on any arbitrary road. Manual collection of ground truth, to build a training model for each individual road, will be cumbersome. Hence we test the unsupervised learning method of K-means clustering on our 6 roads' data. We use classes to cluster evaluation, where class labels of instances in training set are ignored during clustering. After clusters are made, all instances in a cluster are assigned the same class label. In the evaluation phase, assigned class label of each instance is compared with its actual class label, and accuracy values are reported. For Gandhinagar, we obtain 100% accuracy. For Bhandup, Chandivali, Ghatkopar and Powai (Hiranandani), accuracies obtained are between 85.7-94.33%.

For Vikhroli, we obtain accuracy of only 65.52%. This is because the honk based metrics are unsuitable for this road. Using only speed-based metrics, we get 96.55% accuracy. But what metrics are suitable for which road will not be evident without any training. Thus Vikhroli presents a corner case, where our system, without training, will perform poorly.

Temporal variation of traffic

The final question that we seek to answer in this report is what optimizations and enhancements can we make to our system, based on the temporal variation in traffic state at a particular location. We deploy our unit at Bhandup for six days, Dec1-Dec3 and Dec6-Dec8, from 10.30 am to 9:30 pm. On Dec1, we remain at the deployment location from 10:30am-12:30pm and again from 4pm-9:30pm, to observe the ground truth of traffic state. This manual observation is necessary to have an idea of correctness of traffic pattern reported by our unit on the other five days. Our observations on Dec 1 are listed in Table 3.4, where 'F' signifies free-flow and 'C' signifies congestion. Video clips, corresponding to each entry in the table, can be found at [13].

Time	State	Time	State	Time	State
10:30am	F	11:00am	F	11:30am	C
12:00noon	C	12:30pm	F	–	–
4:00pm	F	4:30pm	F	5:00pm	F
5:30pm	C	6:00pm	C	6:30pm	C
7:00pm	C	7:30pm	C	8:00pm	C
8:30pm	C	9:00pm	C	9:30pm	C

Table 3.4. Traffic State at Bhandup on Dec 1, 2010

Our observations from the six days' data, along with their implications in enhancing our system are listed below.

(1) We get meaningful data only between 10:30am-12:30pm and 5:00pm-10:00pm everyday. The other times have hardly any honk detected, indicating the road to be mostly empty. This implies that after identifying the periods when a particular road remains empty, our system on that road can be shut down in those periods. This will save a lot of power.

(2) Some stray minutes can have excessive honks in free-flow due to some temporary reason like a car parking on the road-side. Similarly, congested traffic can have few silent minutes without honks. This implies that, though metric values should continue to be reported to the server every alternate minute for regular updates, the server should look at the data as a time-series, instead of seeing per-minute data in isolation. This will help in removing outliers and take correct decision about traffic state .

(3) During some times of the day, traffic state fluctuates between congested and free-flowing every few minutes. This happens when state is changing from free-flow to congested or vice-versa, as traffic queue buildup and clearance do not happen instantaneously. Traffic is slow at these times. This implies that, time series analysis of per-minute updates at the server will help, where fluctuations over small time intervals can be categorized as a third traffic state intermediate between free-flow and congested states - *slow*.

3.4 Acknowledgement

Without Pankaj Siriah sir, who taught me everything from scratch about hardware design, I would not have anything to show in this APS. Starting from choosing among alternative designs to suit my application, schematic designing in ORCAD to debugging serial ports, the list of things that I learnt from him is endless. He also has a huge contact base that made my buying of hardware components a lot easier. Pramode Mhaske, an associate of Pankaj sir, gave some invaluable tips when I got stuck with using the GPRS modem. I sincerely thank Jayesh Bhai from Cymoline Batteries for patiently listening to my application details and designing battery packs for me. I thank the didis in Rane soldering shop who patiently soldered components on my PCB even at odd hours when I turned up. I thank Deo Comtech for their ready-made boxes that suited my prototype and Pulraj electronics for timely delivery of GPRS modules. My special thanks to Pradeep Raman sir for his compact PCB design. Prof. Dipankar Roy owes special gratitude for suggesting the use of FM to reduce hardware. My sincere thanks to all students of ERTS and ASL lab for humoring me when I bugged them for help in soldering connectors. Last but not the least, was help from Sanjay Bhai in office, who drilled holes in my boxes to fix clamps for deployment.

Implementing bandpass filtering in c5505 ezdsp would not be possible without help from Chitralekha Gupta.

The E2E community of TI, as well as labmates Prashima Sharma and Vijay Gabale, were necessary allies in solving software issues while I programmed my prototype.

Discussions with Devsree Sane helped me decide some interesting machine learning questions to answer using my prototype. Her help was also invaluable in learning open source machine learning tools, input data formatting and interpreting the results.

A huge thanks to the several shopkeeper uncles who let me deploy my units in their premises. My special thanks to the people at Bhandup LG electronics shop, where I spent seven and a half hours on a single day. The number of times they offered me water made me shy at their hospitality.

I thank Prof. Abhay Karandikar and Ajit Jena sir of Computer Center for providing me with a public IP address, so that my unit could talk from outside IITB to my lab machine.

I thank the Parksite police station and Mr. Joseph Reddy for giving me permissions to do on-road work in the several locations.

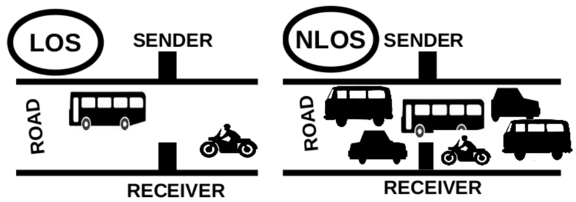
Prof. Bhaskar Raman, Prof. Puru Kulkarni and Sujesha Sudevalaym suggested very useful corrections for my submitted paper. I am also grateful to my advisor, Prof. Bhaskar, for unquestioningly providing me funds in advance, so that I could buy all hardware components. Most importantly, if he had not provided me with the initial idea that speed can be calculated from sound, I would not have a PhD problem at all.

Chapter 4

WirelessAcrossRoad: RF Based Congestion Detection

The basis of our proposed road congestion detection technique is the differential behaviour of RF wireless links in LOS (line-of-sight) vs NLOS (non-LOS) conditions. In the field of wireless networks, there have been several studies [15, 16, 17], to characterize wireless link behavior in different environments. [15] has done a series of controlled experiments using seven 802.11 a/b/g nodes in an indoor office environment. Experiments have been conducted for nodes in clear line of sight (LOS) with each other and also in partial or no line of sight (NLOS). [16] has done similar experiments in three environments - road with foliage, corridor and lab. They have used 802.15.4 compliant Moteiv Tmote Sky motes. Two key results, reported in both these papers are (1) mean Received Signal Strength Indicator (RSSI) in NLOS conditions is much lower than the mean RSSI in LOS. (2) Variance of NLOS RSSI is much higher than variance of LOS RSSI. The signal strength at a wireless receiver degrades due to multipath fading, scattering and reflection. If obstacles hinder LOS between the sender and the receiver, these propagation effects can become more acute. Low value and high variability of RSSI cause related link characteristics like link quality indicator (LQI) and packet reception to suffer too. In conventional wireless and sensor networks, this is considered as a negative phenomenon. Node placement with LOS is preferred to prevent this, and in unavoidable situations of NLOS, protocols try to adapt link parameters to improve network performance.

In this report, we seek to positively exploit this difference in link behaviour between LOS and NLOS conditions. As shown in Fig. 4.1, we place a wireless sender receiver pair across a road. Fig. 4.1(a) shows free-flowing traffic. In this condition, sender and receiver are mostly in LOS, other than when few vehicles pass between them at relatively high speed. Fig. 4.1(b) shows congested traffic. In this condition, sender and receiver are mostly in NLOS, other than when movement of some vehicles create short-lived LOS. In our method, we propose that the sender continuously send packets to the receiver and the receiver log the link metrics like RSSI, LQI and packet reception. Based on the results in [15, 16], we expect these metrics to show marked difference in values between free-flowing (mostly LOS) and congested (mostly NLOS) conditions, which can be used to develop a technique for congestion detection.



(a) Free-flowing traffic (b) Congested traffic

Figure 4.1. Wireless Communication Across Road

Which wireless technology to use is an important design question. Ideally, we want (1) the technology to show marked difference in link characteristics under LOS and NLOS conditions, (2) power consumption to be low, as sensors might need to run on batteries if grid power is unavailable in road deployments, (3) hardware to be inexpensive, (4) minimum 20 m range to be usable across roads and (5) off-the-shelf hardware to be available for the initial proof-of-concept. We initially tried using *Bluetooth* dongles; these links show good difference in link characteristics between LOS and NLOS indoor, but fail to establish link even in LOS on actual roads wider than 10 m. *802.11 Wi-Fi* has high power requirements for both radio and computation platform [33] and hence seems less suitable for this application.

Active *Infrared* (IR), can be used to send condensed beam across road. There is, in fact, a commercial system [34], that counts vehicles based on beam-cutting by vehicle wheels. But this system has been tested only in very sparse lane-based traffic. On roads, which are wide as well as busy, even free-flowing traffic can cause continuous beam-cutting, giving false indication of congestion. Thus RF with its spread propagation model, instead of light with sight propagation, seems more practical to use for this application.

In this report, we have used IEEE 802.15.4 (Zigbee) compliant CROSSBOW Telosb motes with CC2420 radios. These motes consume 35 mW of active power and have a unit price of $80. The outdoor range with integrated antenna is 75-100m [35], which is much more than the width of most Indian roads. Choice of this technology thus satisfies the desired conditions (2), (3), (4) and (5) listed earlier.

To check (1), i.e. performance difference between LOS and NLOS, we performed some preliminary experiments. Using two motes, one as sender and one as receiver, we do similar experiments, as in [16], inside our five floor CSE department building. The sender and the receiver are placed sometimes in LOS and at other times in NLOS and RSSI, LQI and packet reception metrics are measured at the receiver. The key observations from these experiments are that (1) RSSI and LQI variations are very high in NLOS compared to LOS

(2) Absolute value of RSSI and LQI is low in NLOS compared to LOS. (3) Packet loss is much higher in NLOS than in LOS. These results match the results in prior work, giving us confidence to use the motes in our proposed road congestion detection technique.

4.1 Experimental Evaluation

In this section, we describe the various on-road experiments, that we performed to design and evaluate our proposed technique.

4.1.1 Setup

We create the experimental setup shown in Fig. 4.1 on Adi Shankaracharya Marg, a road in Mumbai, 25m wide in each direction. This road has all varieties of vehicles – trucks, buses, cars, autos, bikes and remains heavily congested from approximately 6:45 pm - 8:45 pm on most weekdays. The congestion is because of slow lane merging with another road ahead. We keep the sender and receiver across the road on a line perpendicular to the length of the road, i.e. at a distance of about 25m. The height of the motes from the ground is about 2 feet. The sender sends 25 packets per second of 100 bytes each at -25dBm transmit power. The receiver logs the RSSI and LQI values for each received packet. One person stands on the roadside footpath holding the receiver. The other stands across the road, on the road divider, with the sender. These two persons also observe the road to note the ground truth of the traffic situation. We collect 14 logs of 5 minutes each from about 5:30 pm to 7 pm. The ground truth observed is, initially traffic is free-flowing, it becomes slow at around 6:20 pm and then heavy congestion sets in within a short time and remains so until the end of the experiment. The left side of Fig 4.2 illustrates the free-flowing traffic while the right side shows congestion on this particular road. Videos of the traffic during the experiment can be found at [36].

Figure 4.2. Free-flowing to congested traffic

4.1.2 Method of analysis

An insight that we got from our indoor experiments is that, there is some inherent variability in the wireless link characteristics. This is also reported in prior studies [16]. RSSI can occasionally be good in NLOS and bad in LOS. So instead of isolated values, we think the distribution of link metrics, over some time interval T, will smoothen out such fluctuations and show the average behavior more clearly. A second advantage of using distribution is as follows. Sometimes NLOS is created in free-flowing traffic if a particular vehicle or pedestrian comes and stops in front of either mote. Similarly LOS can be created in congested traffic through gaps between standing vehicles. Such transient fluctuations in the metric values can be smoothened using a distribution. Thus in our road experiments, we plot the CDF of the metrics RSSI, LQI and packet reception rate over a duration of T = 5 minutes. As seen from our results, this handles fluctuations well. We compare the CDF of the metrics between free-flowing and congested traffic to see if they are different.

4.1.3 Results

Figures 4.3, 4.4 and 4.5 show the CDF of RSSI, LQI and packet reception rate respectively. Each graph shows 14 plots: each a CDF calculated over 5 minutes. RSSI and LQI are obtained for only the successfully received packets.

As seen from the figures, the curves in each graph can be classified into three distinct groups – *Group1* between 5:37-6:21pm, *Group2* between 6:22-6:27pm and *Group3* between 6:30-7:05pm. The ground truth of traffic state noted was free-flowing till 6:20pm, slow for a short time and then heavily congested till the end of the experiment. Thus *Group1* corresponds to free-flowing traffic, *Group2* to slow traffic, intermediate between free-flowing and congested and *Group3* to congested traffic. The high correlation of the CDFs with traffic state is apparent visually. For e.g., (a) the 50^{th} and 70^{th} percentiles of RSSI are around -93dBm in congestion and -78dBm in free-flow (b) the 20^{th} and 40^{th} percentiles of LQI are around 85 in congestion and 105 in free-flow (c) the 60^{th} and 80^{th} percentiles of reception rate are around 0 packets/sec in congestion and 24 packets/sec in free-flow. Thus the trend of CDFs corresponding to different traffic states differ widely. This shows that this technique has great promise in traffic state classification.

4.1.4 Choice of experimentation parameters

Transmit power - CC2420 has 8 discrete power levels . The difference between CDF of metric values with LOS and with NLOS remains similar for the different power levels. Hence choice of transmit power does not seem to be critical. We use the lowest power of -25 dBm in our experiments.

Height from the ground - The body of most vehicles, other than buses and trucks, rises to about 3 feet from the ground. Keeping the sender and receiver pair too high will therefore defeat the purpose of creating NLOS under congestion. Keeping them too low will cause multipath reflection with the road, causing instability in link characteristics. Our initial experimental results remain similar for 1.5 feet and 2 feet from the ground. We use a height of 2 feet from the ground in most experiments.

Distance between sender and receiver - The same experiment, as described in Section 4.1.1, is also conducted on a narrow road, approximately 8 m wide. Here link characteristics show no difference between LOS and NLOS. This is because, at such small distance, RSSI is so high that NLOS doesn't have much effect even at the lowest transmit power.

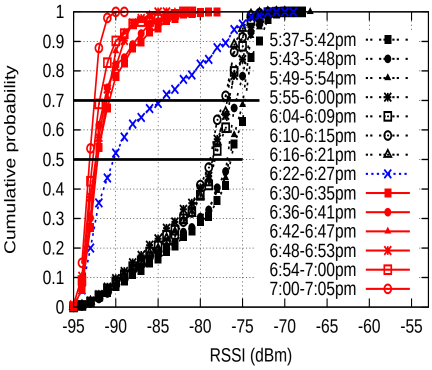


Figure 4.3. CDF of RSSI (dBm)

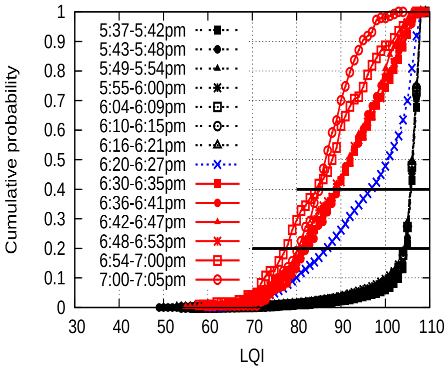


Figure 4.4. CDF of LQI

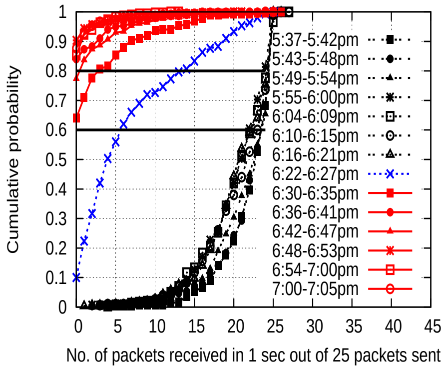


Figure 4.5. CDF of Reception Rate

To be able to use our technique on narrow roads, we thus decide to put the sender in Position 2 instead of Position 1, as shown in Fig. 4.6. Thus though d is small, d' will be large, giving good difference in link characteristics between congested and free-flowing states.

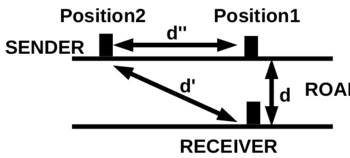


Figure 4.6. Distance between sender and receiver

As directly measuring d' is difficult on a busy road with vehicles passing by, we measure d'', as shown in Fig. 4.6. Keeping the receiver fixed, we moved the sender from d''=5m to d''=40m in steps of 5m and let the receiver log for 5 mins at each sender position. We did this both in free-flowing and congested conditions. The CDF of reception and RSSI, for both traffic states, are shown in Fig. 4.7 and 4.8 respectively. We show the plots only for 15m, 20m, 25m and 30m to prevent the figures from getting cluttered.

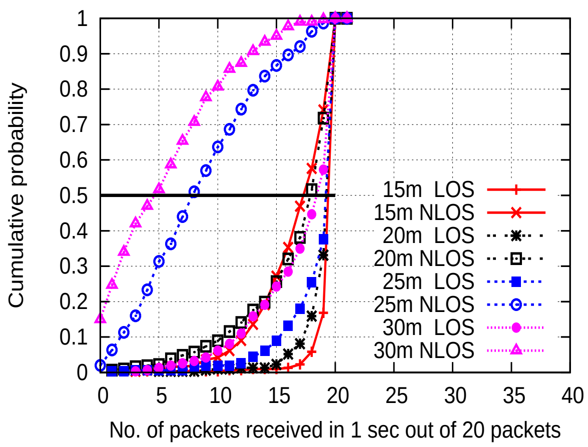


Figure 4.7. CDF of Reception Rate

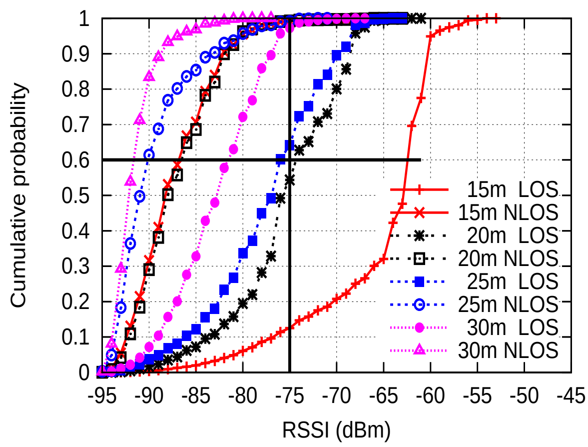


Figure 4.8. CDF of RSSI

In Fig. 4.7, the CDFs in LOS and NLOS are very close for $d'' <= 20m$ and easily distinguishable for $d'' >= 25m$. For eg., 50^{th} percentile reception rate is 8 packets/sec in NLOS and 20 packets/sec in LOS for $d'' = 25m$., while it is about 18 packets/sec in both NLOS and LOS for $d'' = 20m$. Similar results are obtained for CDFs of LQI. Thus any $d'' >= 25m$ seems to be a good choice. When a new unit is placed on an unknown road, the rule of thumb can be to place the sender and receiver in such a way that the 60^{th} percentile RSSI in LOS <= -75dBm. This is marked with black lines in Fig. 4.8. If this condition holds, then all the three link metrics are expected to degrade under NLOS. We make the rule of thumb in terms of RSSI, instead of distance, because RSSI and other link characteristics are a function of not only distance, but environment too.

We repeat the experiment described in Section 4.1.1, on the narrow road, with $d'' = 25m$ for 3 days, collecting about 3 hours log everyday. The CDFs obtained are similar to Fig. 4.3, 4.4 and 4.5 and match with manually observed ground truth. We have thus successfully tested our technique on two kinds of real roads: wide as well as narrow.

4.2 Binary Classification of Traffic States

In this section, we explore how to design a traffic classifier based on our experimental data. The classifier will take as input a 5 minute log from the wireless receiver on the road. It will output a classification of the corresponding 5 mins as having congested or free-flowing

# of *train_test* cases	Decision tree model	Average accuracy (s.d)	# of false positives	# of false negatives
34	lqi_20^{th}percentile <= 94 : 1	97.64%(4.96)	7	1
8	rssi_20^{th}percentile <= -91dBm : 1	90%(0)	0	8
3	rssi_30^{th}percentile <= -89dBm : 1	90%(0)	3	0
3	lqi_20^{th}percentile <= 93 : 1	90%(0)	0	3
2	rssi_40^{th}percentile<= -87dBm : 1	90%(0)	2	0

Table 4.1. Decision tree based classification results

traffic. We do not use a third state of slow traffic, intermediate in between the congested and free-flowing states, as that state rarely occured during our experiments and we have very little data corresponding to it. We use decision tree classifier from WEKA, a widely used open source package for machine learning tools [32], as decision trees are known to perform well on datasets with small number of features.

For the narrow road, we have 9 hours 20 mins of data labeled with manually observed ground truth. This gives us 112 logs of 5 minutes each, 66 of which are labeled as congested and 46 as free-flowing. From each log, we compute 3 CDFs of RSSI, LQI and packet reception rate and from each CDF, we compute the 20^{th}, 30^{th}, ..., 90^{th} percentiles. Thus each 5 minute log gives 3 CDFs, each of which gives 8 percentile values, giving 24 values in all. We create a data set for the 112 logs, with the aforementioned 24 values as features and a class label of 1 for congested and 0 for free-flowing traffic, based on ground truth. On this dataset, we repeat the following *train_test* procedure 50 times — (1) randomly order the instances in the dataset, (2) train a decision tree model using 102 instances from the beginning of the dataset as training set, (3) test the decision tree model using remaining 10 instances as test set. The resulting decision tree model (condition : class), with average accuracy in classifying test data and number of false classifications are given in Table 4.1. We define false positive as predicting an instance to be congested (class 1) while it is actually free-flowing (class 0) and false negative as the converse of this.

As seen from the table, out of the 50 *train_test* cases, the same decision tree model of (lqi_20^{th}percentile <= 94 : 1) is built 34 times. This model has 97.64% classification accuracy with only 8 false classifications. In the remaining 16 *train_test* cases, few other models are built, that have lower accuracy than the first model. Thus a simple threshold based classifier based on the rule (lqi_20^{th}percentile <= 94 : 1) will probably give good classification results for traffic states on this road. But we should validate this in future on a much larger dataset. We get similar classification results for another 5 hours of data collected on the wide road.

4.3 Acknowledgement

Unlike the work using acoustic sensing, this part of the work using RF sensing was done as a group effort. M.Tech student Swanand Kulkarni took part in this work as part of his course project and also as his RA work. Dual degree student Swaroop Roy took part in this work for his final year project. Lokendra Kumar Singh did some work as part of a course project. The individual contributions are listed below.

1) Programming motes to send and receive packets - Lokendra and Swanand, as part of their advanced wireless course project in Jan-Apr sem, 2010
2) Idea that this course project can be extended for on-road sensing - Rijurekha, who was teaching assistant for that course
3) Initial experiments on road using Zigbee motes - Swanand
4) Campus and on-road experiments using Wi-Fi and Bluetooth - Swaroop
5) Designing systematic road experiments to catch traffic pattern change over continuous time in the evening - Rijurekha
6) 30 hours on-road experiments - Swaroop, Rijurekha, Swanand in pairs
7) Finding data analysis method using CDF over time to smooth oscillations in data, after analyzing initial 4-5 hours data - Rijurekha
8) Processing remaining 24 hours of data, plotting graphs - Swaroop
9) Idea that sender receiver have to be placed diagonally for the technique to be used on narrow roads - Swaroop
10) Binary classification of traffic states using decision tree - Rijurekha
11) Writing 6 page workshop paper - Rijurekha
12) Presentation of accepted paper at workshop - Swaroop, Swanand
13) Paper and presentation reviews - Prof. Puru Kulkarni, Prof. Bhaskar Raman

I sincerely thank Swaroop and Swanand for taking a lot of initiatives and making the joint work enjoyable. They deserve special thanks for bearing my holier-than-thou attitude while I worked with them and I'll sincerely try to reduce that as well as my temper, to make team work more effective from now on. My special thanks to Prof. Puru and Prof. Bhaskar for their paper reviews and overall supervision of the work.

Chapter 5

Future Work

5.1 Acoustic sensing

Between Jul-Dec, 2010, I developed a binary traffic state classification technique, comprising of acoustic sensing and data analysis. Between Feb-Aug and Nov, 2011, I built a hardware prototype for real time congestion monitoring based on this technique. I also tested training and non-supervized learning of prototype on 6 Mumbai roads. My two papers *Horn-Ok-Please* and *RoadSoundSense* together have work equivalent to or more than the following papers i.e. developing and testing a sensing technique and preliminary prototyping, deployment and data analysis to see if intended application is supported or not.

1. Surface Street Traffic Estimation University of Michigan, Mobisys'07 (GPS trace based binary traffic state classification)
2. The Pothole Patrol MIT, Mobisys'08 (Accelerometer based road surface monitoring)
3. Nericell MSRI, Sensys'08 (Smartphone based road surface monitoring and congestion detection)
4. Model-Based monitoring of Early Warning Flood Detection MIT, Sensys'08 (2 deployments)
5. Teleport Bell Labs, India, rejected at Sensys'09, but patent filed (Bluetooth based travel time estimation, 4 deployments)

There are several other papers with similar focus, I am referring only to the good ones.

5.1.1 Should I aim a large scale deployment experience paper next?

Two excellent papers on this:

1. "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks", IPSN'07
   - 3 authors are from eecs.berkeley, 3 from ce.berkeley (civil), 1 from crossbow technology
   - focus on hardware design, robustness, power and actual deployment related issues
   - only 0.25 page data analysis out of 10 pages
2. "Air-dropped Sensor Network for Real-time High-fidelity Volcano Monitoring", MobiSys'09
   - 5 authors from Washington State University, 1 author from U.S. Geological Survey
   - the same author list, reordered, have four papers listed in references. 1 paper on sensing hardware design, 1 paper on reliable data dissemination protocol, 1 paper on a TDMA MAC, 1 paper on lightweight network management, these papers are over 3 years, not in Tier-1 conferences, Tier-2 or workshop
   - focus on assimilating several prior works and building a system, deploying it and hardware/software lessons learnt while deploying
   - only 0.5 page data analysis out of 14 pages

Some common features of both papers are as follows.

1. Have big team comprising of people with diverse skill sets.
2. Have gradually built their systems over 3-4 years before large scale deployment, both papers have at least 4 self references of several components that these papers have stitched together.
3. They have not cared about the data analysis from application perspective i.e. Whether there will be volcanic eruption or whether Golden gate bridge will collapse; they are given some requirements of data, periodicity, scale of deployment and they are trying to meet those. Why are the requirements like that, they don't bother.

The requirements which I need to meet, if I want a large scale deployment are tabulated in Fig. 5.1. But the question remains that can we write a paper after all this? I mean there are already some good deployment experience papers, what new story would we have? We also don't have a sensor network, unlike the above two papers. Ours is a star topology. Increase in scale means replication adding to logistic and management overhead, not any network issue. A possible solution might be as follows.

Category	To-do	Remarks
Technical	**Make a second more robust prototype (custom board, better connectors, better packaging)**	**Help from Pankaj Sir (CSE), Pramod Sir (non IIT), Akshay, Nikhil, Dipankar Sir (EE), Rajesh Sir (ASL), Swapnil (Civil) – none of them has any stake in this work, purely “thank you” based model, less effective than collaboration**
Technical	**Power supply**	**Take grid power/ solar**
Technical	**Power optimization**	**Sleep mode, duty cycling**
Technical	**Remote programming of hardware, debugging, more reconfigurability**	**Will an internship in Bell Labs in their Bluetooth based road monitoring project help me learn these skills? Krishna in MSR has a Sensys 2004 paper on structural health monitoring with Ramesh Govindan, Deborah Estrin and Deepak Ganesan – will an MSR internship with him help?**
Logistic	**Finding good locations**	**Hiring project staff, police support**
Logistic	**Getting permission to put up on lamp-posts, draw power**	**Abhay Karandikar has contacts with police?**
Logistic	**Building 50+ units – getting all components, soldering, making connectors, managing 50+ Vodafone connections, hardware-software issues**	**Hiring project staff**

Figure 5.1. To-dos for large scale deployment of acoustic technique

5.1.2 Data Analysis and Algorithms

If the large scale deployment runs smoothly and we have lots of data that we can rely on as correct data, then we can move in directions of the following two papers.

1. A Macroscope in the Redwoods, SenSys’05
    - 4 authors from eecs.berkeley, 4 authors from biology department, berkeley, 1 author from Intel Research, Berkeley
    - 3 pages hardware, software design and deployment, mainly reusing own prior work
    - 8 pages data analysis, probably done by the biologists
2. Vtrack, SenSys’08
    - 5 authors from MIT CSAIL, 1 author from Tel-Aviv University, 1 author from U. Illinois, Chicago
    - No hardware, software design or deployment, uses already available GPS and Wi-Fi traces and builds lots of algorithms for travel time estimation

So only large scale deployment will probably not give another paper. The deployment has to be robust and reliable, so that analysis of data and algorithms can be developed. In our case interesting questions might be finding temporal and spatial patterns in congestion, finding regular hotspots, correlating congestion to travel time by using some parallel technique to measure travel time etc. Collaboration with Civil/ Machine Learning/ Time Series Analysis/ Algorithms people might help.

5.2 RF sensing

Between Sep-Oct, 2010, I, with two other students Swaroop and Swanand, have done very preliminary work to detect congestion using a single pair of zigbee motes across road. But the simplicity of the method and the high accuracy of the results are encouraging. If we can satisfactorily answer a few uncomfortable questions like motivation of using RF instead of more LOS sensitive Infrared sensing and whether our setup will interfere with other ISM band communication on road, then there is a lot of scope of future work in this thread.

5.2.1 Queuelength estimation using array of tx-rx pairs

Application 1 – real time traffic signal time-set based on queue length information from 3 or 4 cross-roads.

Using array of tx-rx pairs across road, as shown in Fig. 5.2, whether traffic light times can be set in real time can be tried. A list of to-dos, if this has to be done are as follows.

1. Is the application important? Quantify loss due to improper signal time-setting at 20 Mumbai junctions. Manually see if real-time signal-time setting will cause any improvement.
2. Make the work done more rigorous decrease time required to detect congestion, identify the best metrics, program the mote so that receiver itself can take classification decision based on logs instead of offline analysis.
3. Strict time-synchronization among all motes (FTSP, GPS).
4. I don’t think any new MAC or routing will be needed, as everything can be statically decided. Still explore if needed.
5. Hardware design Sink node (R) at traffic signal to have flash and GPRS to feed data to Tom Mathew’s simulator.

6. 2.4 GHz at 2 feet is ok for 25 m across road, but probably not for 100m along road. MHz radio is suitable? Commercially available? MHz band free in India? Dual radio integration on mote? Should we use relay nodes instead? What about using RF for sensing and ultrasonic/FM for communication?

7. Building system after making 3)-7) work, test, debug thoroughly in lab. On road testing, feeding data to Tom Mathew's simulator and analyze results.

8. Duty cycling, triggered sensing.

9. System cost and power analysis.

Possible Publication - Sensys, Apr 2012.

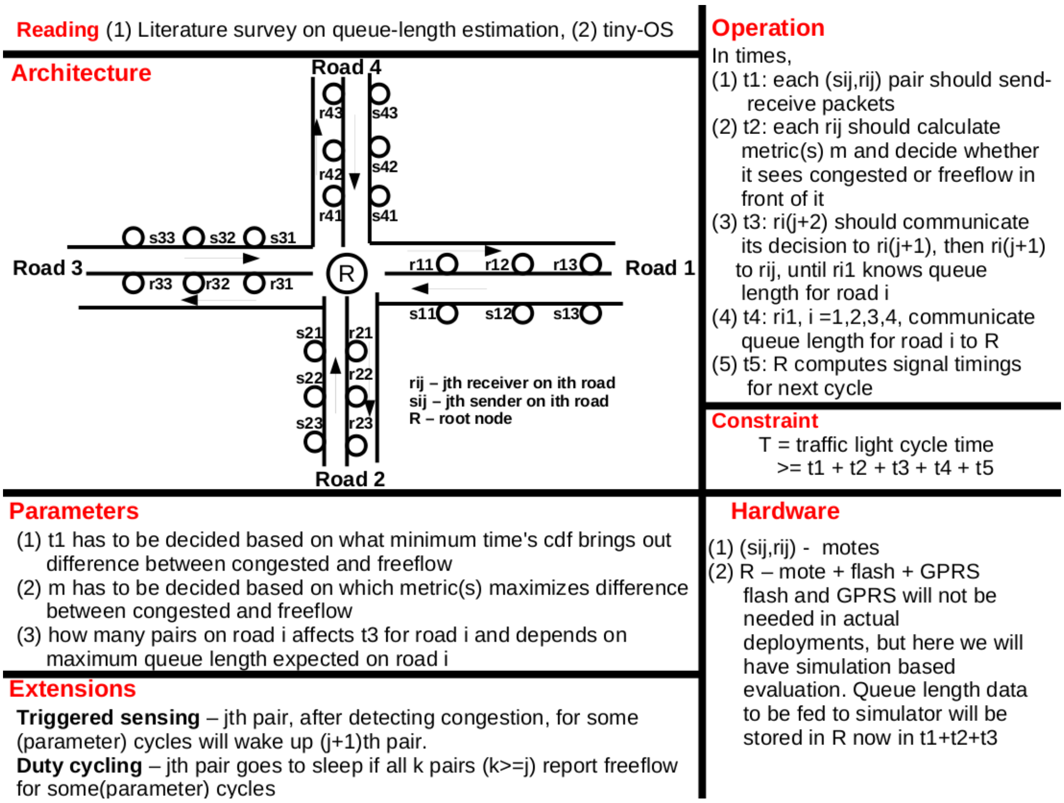


Figure 5.2. Traffic light control architecture

Application 2 – Correlating travel time to queuelength, predicting bus arrival time based on travel time estimates

A list of to-dos,if this has to be done are as follows.

1. Use Akshay/Ninad's GPS hardware on taxi-fleets and try to correlate travel time with queuelength. Akshay is Abhay Karandikar's PhD student and Ninad is a project staff in Systems and Controls.

2. Rigorous testing on many Mumbai roads to see whether even without GPS in vehicles, just using road side infrastructure, we can estimate travel time for chaotic traffic

3. Extend Prof. Abhiram Ranade's Mumbai Navigator with bus arrival time predictions instead of Poisson arrival of buses at bus-stop after solving associated algorithmic issues.

Possible Publication Mobisys, Dec 2012

5.2.2 Multilevel traffic state classification

Instead of classifying traffic into two states of congested and free-flowing, can we have several states like empty road, very fast traffic, fast traffic, slow traffic, very slow traffic, stagnant traffic? A list of to-dos,if this has to be done are as follows.

1. Setting up infrastructure for imaging

2. Setting up infrastructure to collect receiver logs

3. Ascertaining ground truth for multilevel classification (image processing/ manual judgment)

4. Correlating logs to ground truth

5. Large scale on road experimentation

Possible Publication Secon, Dec 2011

Bibliography

[1] http://www.nhai.org/roadnetwork.htm.

[2] http://www.neoncarrot.co.uk/h_aboutindia/india_statistics_1.html.

[3] http://en.wikipedia.org/wiki/Infrastructure_in_Bangalore.

[4] Benjamin Coifman and Michael Cassidy. Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice*, 36(10):899–917, 2002.

[5] Li Li, Long Chen, Xiaofei Huang, and Jian Huang. A Traffic Congestion Estimation Approach from Video Using Time-Spatial Imagery. In *ICINIS*, 2008.

[6] Sing Y. Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. Paper ucb-its-pwp-2004-7, California Partners for Advanced Transit and Highways (PATH), 2004.

[7] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface street traffic estimation. In *MobiSys*. ACM, 2007.

[8] http://traffic.berkeley.edu/theproject.html.

[9] Katrina LaCurts Samuel Madden Hari Balakrishnan Sivan Toledo Arvind Thiagarajan, Lenin Ravindranath and Jakob Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Sensys*, November 2009.

[10] Video of a road intersection in bangalore, india. http://research.microsoft.com/en-us/um/india/groups/mns/projects/nericell/index.htm.

[11] Chaotic traffic: representative videos. http://www.cse.iitb.ac.in/~br/horn-ok-please/.

[12] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*. ACM, 2008.

[13] http://www.cse.iitb.ac.in/~riju/rss-videos.

[14] Rijurekha Sen, Bhaskaran Raman, and Prashima Sharma. Horn-ok-please. In *Mobisys*, June 2010.

[15] Arunesh Mishra Suman Banerjee Vivek Shrivastava, Dheeraj Agrawal. Understanding the limitations of transmit power control for indoor wlans. In *IMC*, October 2007.

[16] Naveen Madabhushi Dattatraya Y Gokhale Phani K Valiveti Bhaskaran Raman, Kameswari Chebrolu and Dheeraj Jain. Implications of link range and (in)stability on sensor network architecture. In *WiNTECH*, September 2006.

[17] Ramesh Govindan Jerry Zhao. Understanding packet delivery performance in dense wireless sensor networks. In *Sensys*, November 2003.

[18] http://www.techtransfer.berkeley.edu/newsletter/08-3/vehicle-detection-with-wireless-sensors.php.

[19] http://www.financialexpress.com/news/by-next-year-well-target-10-of-indias\\-smartphone-market/551198/0/.

[20] S. Bhattacharyya D. Banerjee, S. K. Chakraborty and A. Gangopadhyay. Evaluation and analysis of road traffic noise in asansol: An industrial town of eastern india. In *International Journal of Environmental Research and Public Health*, Sept 2008.

[21] S. Agarwal and B.L. Swami. Road traffic noise annoyance in jaipur city. In *International Journal of Engineering Studies Volume 1, Number 1, pp. 3946*, 2009.

[22] Abirlal Mukherjee Biva Roy Debashis Chakrabarty, Subhas Chandra Santra and Prasun Das. Status of road traffic noise in calcutta metropolis, india. In *Acoustical Society of America*, 1997.

[23] Brahma D. Tripathi Vinita Pathak and Virendra Kumar Mishra. Dynamics of traffic noise in a tropical city varanasi and its abatement through vegetation. In *Environ Monit Assess: Springer Science + Business Media*, 2007.

[24] http://www.made-in-china.com/showroom/rosstone/product-detailioZJEvnDJmVW/China-VHF-Wireless-Tie-Clip html.

[25] http://www.ahujaradios.com/Productdetail.asp?prd_id=128&id=6&sc_id=32.

[26] http://focus.ti.com/docs/toolsw/folders/print/tmdx5505ezdsp.html.

[27] http://wm.sim.com/Sim/FrontShow_en/wireless/detail.aspx?cid=6&nid=36.

[28] http://in.mouser.com/search/ProductDetail.aspx?qs=boflksbqthoLObwWTE7%252b%252bw==.

[29] http://www.datasheetdir.com/AP1512AT5+Step-down.

[30] http://www.deocomtech.com/water_proof.htm.

[31] http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html?DCMP=dsp_ccs_v4&HQS=Other+OT+ccs.

[32] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software: An update. In *SIGKDD Explorations, Volume 11, Issue 1*, 2009.

[33] Kameswari Chebrolu Vijay Gabale, Bhaskaran Raman and Purushottam Kulkarni. Lit mac: Addressing the challenges of effective voice communication in a low cost, low power wireless mesh network. In *ACM DEV*, December 2010.

[34] http://www.ceos.com.au/products/tirtl.htm.

[35] www.willow.co.uk/TelosB_Datasheet.pdf.

[36] http://www.cse.iitb.ac.in/~puru/wirelessacrossroad.