Kyun Queue: A Sensor Network System To Monitor Road Traffic Queues

Rijurekha Sen, Abhinav Maurya, Bhaskaran Raman, Rupesh Mehta,
Ramakrishnan Kalyanaraman, Nagamanoj Vankadhara, Swaroop Roy, Prashima Sharma
{riju, ahmaurya, br, rupeshm, krishnan, nagamanojv}@cse.iitb.ac.in,
{bswarooproy, prashimasharma}@gmail.com
Indian Institute of Technology, Bombay

Abstract

Unprecedented rate of growth in the number of vehicles has resulted in acute road congestion problems worldwide. Better traffic flow management, based on enhanced traffic monitoring, is being tried by city authorities. In many developing countries, the situation is worse because of greater skew in growth of traffic vs the road infrastructure. Further, the existing traffic monitoring techniques perform poorly in the chaotic non-lane based traffic here.

In this paper, we present *Kyun*[1] Queue, a sensor network system for real time traffic queue monitoring. Compared to existing systems, it has several advantages: it (a) works in chaotic traffic, (b) does not interrupt traffic flow during its installation and maintenance and (c) incurs low cost. Our contributions in this paper are four-fold. (1) We propose a new mechanism to sense road occupancy based on variation in RF link characteristics, when line of sight between a transmitter-receiver pair is obstructed. (2) We design algorithms to classify traffic states into congested or free-flowing at time scales of 20 seconds with above 90% accuracy. (3) We design and implement the embedded platforms needed to do the sensing, computation and communication to form a network of sensors. This network can correlate the traffic state classification decisions of individual sensors, to detect multiple levels of traffic congestion or traffic queue length on a given stretch of road, in real time. (4) Deployment of our system on a Mumbai road, after careful consideration of issues like localization and interference, gives correct estimates of traffic queue lengths, validated against 9 hours of image-based ground truth. Our system can provide input to several traffic management applications like traffic light control, incident detection, and congestion monitoring.

[1] *kyun*, pronounced as 'few' starting with 'k' instead of 'f', means 'why' in Hindi

SenSys'12, November 6–9, 2012, Toronto, ON, Canada.

1 Introduction

The problem of road congestion is currently plaguing most cities in the world. Long traffic queues at signalized intersections cause unpredictable travel times and fuel inefficiency [1, 2]. The problem is felt more acutely in growing economies like India, primarily because infrastructure growth is slow compared to the growth in vehicles due to space and cost constraints.

Secondly, automated traffic management solutions, developed for western traffic, are at best an accidental fit for roads in developing countries. Traffic in many developing regions is distinctly different from western roads in two ways (see Fig. 1): it is (1) non-lane based and (2) highly heterogeneous. Four wheeler heavy vehicles like buses and trucks, four wheeler light vehicles like cars, three wheeler auto-rickshaws and two-wheeler motorcycles ply the same road, intermingled with each other, without any lane discipline [3].[2]

An ideal traffic sensing system for developing regions has the following stringent requirements: (1) it should sense road occupancy even if traffic is chaotic and non-lane based, (2) should be deployable without interrupting traffic flow, (3) be capable of real time sensing and classification to support applications like traffic signal control and (4) should have low installation and maintenance costs. Section 2 discusses the shortcomings of existing sensing systems like magnetic loops, camera, infrared sensors, acoustic sensors and probe sensors, vis-a-vis these requirements.

In this paper, we present *Kyun* Queue, a sensor network system for real time traffic queue monitoring. We build a new mechanism to sense road occupancy, based on how RF link quality suffers in absence of line of sight. Our system comprises of an IEEE 802.15.4 transmitter-receiver pair across the road, where the transmitter continuously sends packets and the receiver measures metrics like signal strength and packet reception ratio. We show that these metrics show a strong correlation with the occupancy level on the road between them. We investigate which wireless link characteristics, what time windows and what algorithms give a highly accurate real time classification of traffic states. Our system gives above 90% binary classification accuracy on 16 hours of data from two roads in Mumbai.

[2] [3] has several representative videos of chaotic traffic at http://www.cse.iitb.ac.in/~riju/rss-videos/

Figure 1: Chaotic traffic (source [4])

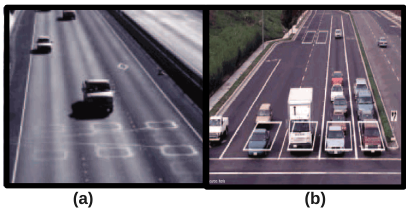

Figure 2: Lane based traffic

Figure 3: Issues with infrared

Building on this binary classification of traffic states using one pair of sensors, we construct a linear array of such sensors. This can detect the length of a queue or the degree of congestion on a given road stretch. We design and build appropriate embedded platforms to do (1) sensing and computation for binary classification, (2) communication to combine the decisions of multiple sensors to know the length of traffic queue and (3) communication of the queue length values to a remote server.

Our system, deployed on a Mumbai road for 9 hours, periodically (every 30 secs in our deployment) reports traffic queue length values to a remote server. Manual examination of the corresponding 9 hours of image-based ground truth, gives maximum accuracy of our queue estimates to be 96% and minimum accuracy to be 74%. These queue estimates can be potentially used in a range of applications such as: automating traffic light control, detecting bottlenecks or creating congestion maps of a city.

The rest of the paper is organized as follows. Section 2 discusses the existing traffic sensing solutions. Our proposed sensing technique and the real time traffic state classification algorithms based on it, are discussed in Section 3. The design of our sensor network is detailed in Section 4 and hardware prototypes are described in Section 5. We present the deployment based evaluation of our system in Section 6. Finally we discuss the issues of sensor localization, interference and power consumption, before concluding the paper.

2 Related work

Automated traffic management solutions have been deployed in many developed countries since several years. In this section, we discuss the traffic sensing methods primarily used to provide input to the traffic management systems and their usability in chaotic traffic.

Loop detectors - There are a lot of research projects [5, 7] and several deployed systems [6] that do vehicle counting using magnetic loops under the road. A vehicle loop detector costs $700 for a loop, $2500 for a controller, $5000 for a controller cabinet and 10% of the original installation cost for annual maintenance [1]. Loops need to be placed under the road. Digging up roads to install and maintain the infrastructure intrudes into traffic flow. Also re-laying the road surface needs re-laying the sensors. Moreover, loops have been traditionally placed under each lane as seen in Fig. 2(a). How should the placement be in absence of lanes is yet to be explored.

In comparison, our sensors are to be placed on roadside, without affecting the flow of traffic. We can sense road occupancy even if traffic is chaotic and non-lane based. Our technique is cheap, with each sensor pair costing about $200. This is the cost of prototype using off-the-shelf modules, which can further reduce with customized design. Each pair gives a binary classification of traffic state as free-flow or congested. To know length of queue, we need an array of sensor pairs. We can cover about 200 meters of road at about $1200.

Image sensors - Video surveillance based traffic monitoring is fairly common [8]. [17] gives a comprehensive survey of the major computer vision techniques used in traffic applications. But the traditional setting for which vision algorithms exist can be seen in Fig. 2(b). For usability in developing countries, algorithms are needed for scenarios like Fig. 1. [18] is a preliminary work on image processing algorithms for chaotic traffic sensing. The algorithms are offline, so the trade-off between computation and communication is not yet understood. Also the sensing accuracy itself has been tested on only 2 minutes of video clip. [19] is another recent work to use low quality images from CCTV for traffic sensing. But computational overhead, real-timeliness and accuracy of the designed algorithms are yet to be evaluated. Thus though image-based sensing shows promise, with advantages such as ease of deployment and potential low cost, there are several aspects that need careful evaluation and validation, especially in chaotic traffic conditions.

In comparison, we present an alternative traffic sensing system that works in chaotic non-lane based traffic. We have thoroughly evaluated the sensing accuracy on 16 hours of traffic in Mumbai. The algorithms are very low overhead and we have implemented them on low end embedded sensor platforms. Sensing and computation are done on the road. Only the traffic queue length values are communicated to the remote server, avoiding the communication overhead of video transfer.

Infrared sensors - Active infrared sensors placed across the road suffer beam cuts by vehicular movement in between. [10] is a commercial product that measures speed, length and lanes of vehicles from beam cuts. However, infrared propagates as a ray, and hence is strictly line of sight (LOS) based. This makes it overly sensitive to even small obstacles. When we tried using it on Mumbai roads, pedestrians who frequently use the roads (Fig. 3(a)) or irregularities of the road surface (Fig. 3(b)) would not allow the tx-rx pair to communicate. Also [10] has been tested only in lane-based traffic. Beam cuts in high density, non-lane based and

Technique	Drawbacks
loop detectors [5, 6], magnetic sensors [7]	high cost, under-the-road installation, do not currently support chaotic traffic
video and images [8, 9]	attractive option for exploration, but no proven solutions for chaotic traffic thus far potentially high computation and/or communication overhead for chaotic traffic
IR sensors [10]	pedestrians, road surface irregularities and non-lane based vehicular movement cause spurious beam cuts
acoustic sensors [11, 12, 3]	high training overhead, slow response
probe sensors [1, 13, 14, 15, 16]	useful for complementary applications, but too sparse and noisy to allow micro management of road networks

Table 1: Drawbacks of existing techniques for chaotic traffic management

heterogeneous traffic as seen in Fig. 1, would need careful study to be used for chaotic traffic measurement.

The setup of sensor pair across road in our system uses 802.15.4 radios which have a spread propagation model, instead of ray propagation model of infrared. This makes our technique robust to noise and thus suitable for disorderly road conditions such as in Fig. 1. This robustness is evident from the rigorous empirical evaluation of our binary traffic state classification over 16 hours of chaotic traffic data. We also enhance the pairwise sensing to an array of sensors and detect length of queues, an aspect not examined in the IR-based systems.

Acoustic sensors - Some recent research is being done to use acoustic sensors for traffic state estimation, especially in developing regions, where traffic being chaotic is noisy [12]. But standing traffic does not have any uniform sound signature and depends largely on the driver behavior and type of vehicles. For example, if vehicles are standing in a long queue awaiting a green signal, many drivers might just shut down the engines and wait silently or might blow honk impatiently, producing two very different sound signatures for same traffic state. This reduces the sensing accuracy [3]. Also, to attain a valid signature correlating to a traffic state, sensing time window of at least a minute or so is needed, making these systems slow in response.

In comparison, our technique works in the order of 10-20 seconds. The classification accuracy is also much higher than acoustic, as vehicles being heavy metallic obstacles always affect RF signal quality. This high accuracy also makes it possible, to take the binary classification to a multilevel classification using an array of RF sensors to know length of queue on a road. Achieving this with low accuracy acoustic sensors would be tricky.

Probe sensors - Significant research is being done to leverage cell phones and participatory sensing to generate traffic related information. Focus has been given to research both in networking issues [20, 21, 22] and machine learning issues [23, 1, 15]. But participatory sensing data is inherently noisy [24]. Also probe vehicles might not be present at a given intersection at all times. Travel time estimates, transit vehicle information and congestion maps to be disseminated to commuters for route selection, can tolerate aperiodicity and noise. Such commuter applications are thus highly suitable using probe data.

Our work is aimed at providing strictly periodic, accurate input to traffic management systems. Our system, based on static sensors, would potentially be used at some key intersections of a city, where intelligent and adaptive control would positively affect traffic flow. This is orthogonal and complementary to the commuter applications handled by probe sensing.

The drawbacks of existing techniques are summarized in Table 1. Some techniques like loops and acoustic sensing have clear disadvantages. Some like probe sensors are ill-suited for the particular applications that we are targeting in this paper. Some others like image and infrared sensing might work after adaptation for chaotic traffic, but there is no working adaptation till date.

3 Traffic sensing with wireless across road

In this paper, we wish to design a system for traffic queue length measurement in chaotic road conditions. This has several potential applications like traffic light control, incident detection, and congestion monitoring. An important point to note here is that, at almost all intersections in developing countries, a variety of vehicles stand as a coagulated mass waiting for green signal (see Fig. 1). On getting green signal, they all move forward almost bumper to bumper, with little variability in speeds. So queue length is proportional to traffic volume, the road width being the proportionality constant. We thus assume, as is intuitive, that finer granularity information like exact vehicle counts are unnecessary. To tune traffic lights proportional to waiting traffic volumes, queue lengths are sufficient. Thus we aim to detect road occupancy in chaotic traffic, and build upon that to detect length of queues.

It is well known in the wireless networking domain, that characteristics of wireless links like 802.11 and 802.15.4 are affected in the absence of LOS [25]. Obstacles cause reflection, absorption and scattering of the RF signal, degrading the link quality. Researchers have exploited these vagaries of RF links in different kinds of application scenarios, the most common being that for indoor localization [26, 27, 28, 29].

The effects of vehicular traffic on wireless links were briefly studied in [30], where the authors seeked to quantify the link quality in harsh deployment scenarios, like tunnels

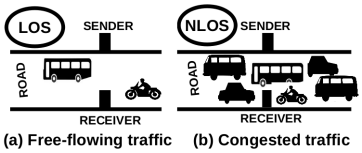

Figure 4: Wireless communication across road

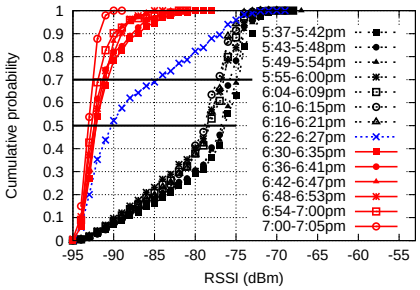

Figure 5: CDF of RSSI (dBm)

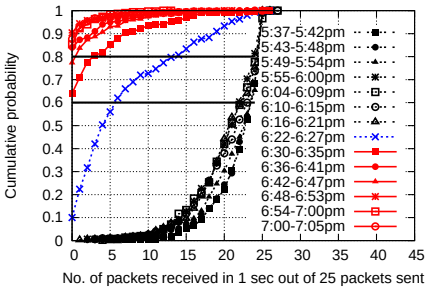

Figure 6: CDF of PRR

with traffic moving through them. In this paper, we want to apply this effect of traffic on wireless links for a completely novel application, to measure road traffic density. The basic intuition is as follows. If a wireless transmitter-receiver pair is kept across a road and made to communicate, the link characteristics, observed at the receiver, should be affected by the vehicles on the road. But unlike IR, these wireless signals follow the spread model of propagation, and hence should not be overly sensitive to small obstacles. Can traffic states be quantitatively inferred from RF link characteristics? Can classifications be done in order of tens of seconds to support applications like traffic light control? What wireless features and algorithms can be used for high classification accuracy? In this section, we seek to answer these questions.

3.1 Does road traffic affect wireless links?

To answer this, we start with some proof-of-concept experiments. We create a setup shown in Fig. 4 on Adi Shankaracharya Marg, a road in Mumbai, about 25m wide in each direction. We keep two 802.15.4 compliant Telosb motes across the road, one as transmitter (tx) and the other as receiver (rx), on a line perpendicular to the length of the road. The tx sends 25 packets per second, each having a payload of 100 bytes, at -25dBm transmit power. The rx logs the number of packets received and Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) for each received packet. One person stands on the roadside footpath holding the rx and another stands across the road, on the road divider, with the tx, both rx and tx being at a height of about 0.5 m from the ground. 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.

Figures 5 and 6 show the CDF of RSSI and packet reception rate respectively. Each graph shows 14 plots: each a CDF calculated over 5 minutes. 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 is free-flowing till 6:20pm, slow for about 5 minutes and then heavily congested till the end of the experiment. Thus *Group1* corresponds to free-flowing traffic, *Group2* to slowly moving traffic, intermediate between free-flowing and congested and *Group3* to heavily 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 60^{th} and 80^{th} percentiles of reception rate are around 0 packets/sec in congestion and 24 packets/sec in free-flow. We see similar trends in 16 hours of data collected over three weeks, from two Mumbai roads. Sample videos of free-flowing and congested traffic are available at [31].

Figure 10: Mismatch of sensing and traffic state

3.2 Real-time classification of traffic states

As seen above, wireless logs of 5 minutes duration show good visual difference in the distributions of wireless characteristics, between free-flowing and congested traffic. Applications like congestion maps and bottleneck detection can be handled at a time scale as large as 5 minutes, but adaptive traffic light control would intuitively need faster inputs. From our observations of Mumbai and Bengaluru traffic lights, traffic signal cycles typically last for about a minute. This one minute cycle time is divided into slots, in which different contending flows get their respective green times. Green time for any flow lasts for about 10-30 secs, though it can go over a minute for critical flows.

Any system like ours, aiming to provide traffic state information to traffic lights, would need an input parameter about the frequency at which traffic queue estimates are needed. We wish to determine the lowest classification time window at which a sufficiently high accuracy (say 90%) can be obtained. Very low time windows give noisy predictions. This noise comes from two sources - (a) the inherent stochastic nature of wireless links which causes link quality to be intermittently bad though the tx-rx are in perfect line of sight (b) the instantaneous traffic condition between the tx-rx are contrary to the actual traffic state. For e.g. tx-rx may be in line of sight between several standing vehicles in congestion (Fig. 10(a)) or several heavy vehicles may pass the tx-rx pair simultaneously in free-flow obstructing line of sight

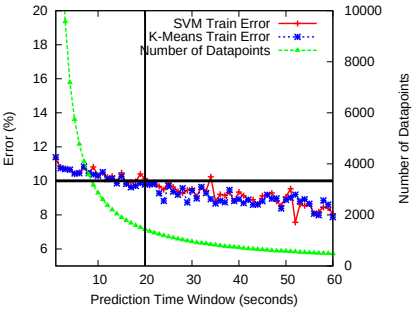

Figure 7: FC algorithm training error

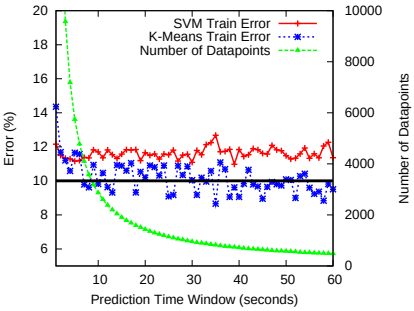

Figure 8: SC algorithm training error

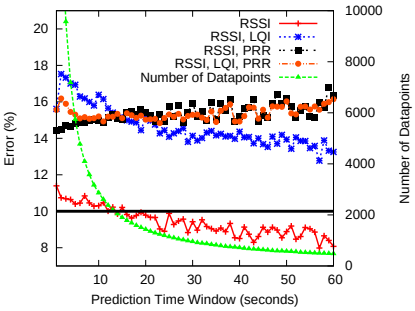

Figure 9: Choice of features

(Fig. 10(b)). Thus we need to choose the classification time window, henceforth referred to as t, carefully.

3.3 Labeled data-set for evaluation

To evaluate our choices of t, features and algorithms for real time traffic state classification, we need a data-set with labeled ground truth. For this purpose, we use the 16 hours data-set, collected using the setup in Fig. 4. The data is collected from two Mumbai roads, a 25 m wide Adi Shankaracharya Marg, henceforth referred to as wide-road and another road, 8 m in width, henceforth referred to as narrow road. Specifically we have 13676 secs of wide-road data labeled as free-flow and 14992 secs labeled as congested. Similarly, we have 13486 secs of narrow-road data labeled as free-flow and 16678 secs labeled as congested. The labeling was done at a larger time-scale of 5 minutes to reduce manual overhead and all the one second long windows, belonging to the same 5 minute window were uniformly labeled. The roads and the times of day of data collection were chosen in a way that traffic states did not toggle within 5 minutes. Thus the error in ground truth observation, even if present, is very small. Representative videos, showing free-flow and congested traffic on the wide road, can be found at [31].

3.4 Classification algorithms

We use machine learning algorithms to do traffic state classification. In this paper, we concentrate on *binary* traffic state classification: *congested traffic*, when vehicles have to brake and stop vs *free-flowing traffic*, when vehicles move according to the driver's intended speed, bounded by the road speed limit. We show that this binary classification is sufficient for queue length estimation, using binary decisions from a linear array of multiple sensor pairs.

To decide what binary classifier to use, the trade-off is between (1) accuracy of classification, (2) implementability on a low end embedded platform, (3) complexity of the classifier models and (4) overhead of model training. Linear hyperplane classifiers are simple enough to implement on our platform (TI's C5505) and to train and test in near real time. SVM and K-Means-based classifiers belong to this category and are state-of-the-art supervized and unsupervized learning algorithms respectively. K-Means, being unsupervized, has the additional advantage of minimal manual labeling of training data. We build four possible algorithms based on these two classifiers and subsequently choose one based on accuracy and labeling overhead.

FeatureClassifier (FC) algorithm - In this algorithm, the wireless data for a certain t is transformed into a feature vector comprising 9 features: from the set of RSSI values of packets received in a time window, the nine percentile values corresponding to 10^{th}, 20^{th}, ..., 90^{th} percentile are drawn as features. The ground truth for the corresponding time window is appended to the generated feature vector. If no packet is received in a time window, a dummy packet having RSSI of -95 dBm, close to the radio sensitivity level, is considered to have been received. The collection of all data points thus obtained comprises the training dataset. This is used to train classification models either using SVM or K-Means. In the testing phase, similar feature vectors are created from wireless data over the same time window t. Then each t is labeled as free-flow or congested based on the training model.

SignalClassifier (SC) algorithm - This meta-classifier alternative to FeatureClassifier, uses majority voting on per-packet congestion predictions. In the training phase, we obtain a per-packet classifier using K-means or SVM, by considering only the packet RSSI as a feature. In the testing phase, for each t, we employ the per-packet classifier obtained in the training phase on each received packet, obtaining a label for each packet. Consider $count^{(t)}_{congestion}$ and $count^{(t)}_{freeflow}$ to be the number of packet-level congestion and free-flow predictions in a time slot t. We predict congestion in the time slot if $count^{(t)}_{congestion}$ is greater than or equal to $count^{(t)}_{freeflow}$ and free-flow otherwise. We evaluate the four algorithms:1) FeatureClassifier using SVM, 2) FeatureClassifier using K-Means, 3) SignalClassifier using SVM, and 4) SignalClassifier using K-Means, on our labeled dataset. Fig. 7 and Fig. 8 show the training errors of the FeatureClassifier and the SignalClassifier algorithms respectively. As we can see, FeatureClassifier using K-Means outperforms the other algorithms in terms of accuracy. Also, the classification error of SignalClassifier is not as well behaved as FeatureClassifier, and is surprisingly higher for the supervized SVM algorithm than the unsupervized K-Means algorithm. This is due to the increased label noise in translating the ground truth of a 5 minute time window to each packet received in that window.

The accuracy for FeatureClassifier using K-Means is

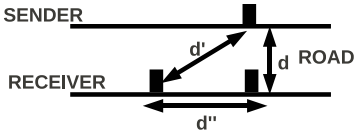

Figure 11: Sensing on narrow road

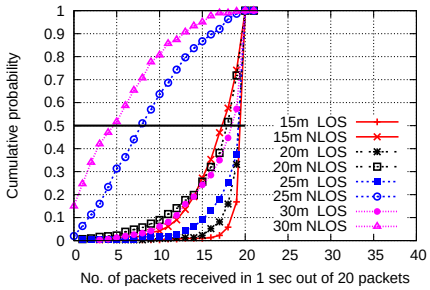

Figure 12: CDF of PRR with d" variation

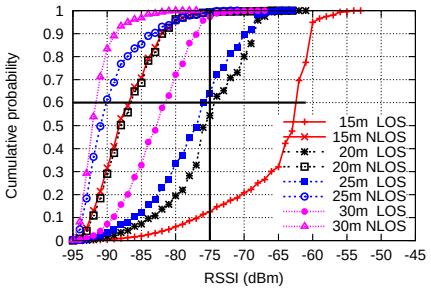

Figure 13: CDF of RSSI with d" variation

above 90%, when the classification time window is at least 20 seconds, shown by a vertical line in Fig. 7. Hence, we choose FeatureClassifier using K-Means and t as 20 seconds in our *Kyun* Queue system. The graphs presented here are for the wide-road dataset, but we observed similar results for the narrow-road dataset as well.

3.5 Choice of features

RSSI, LQI and Packet Reception Rate (PRR), all showed good visual difference in distributions between free-flowing and congested road traffic. But there are several IEEE 802.15.4 compliant radios like XBEE[3], which do not report LQI values. Thus we seek to understand the effect of not using LQI for traffic classification. Using the wide-road data, we plot the training error of an SVM classifier in Fig. 9. Interestingly, RSSI percentile-based features yield better accuracies than using additional percentile features based on LQI and PRR. The reason for this non-intuitive observation is that RSSI is much more strongly correlated with line-of-sight than LQI or PRR.

This strong correlation of RSSI with line-of-sight is also evident from an experiment that we perform on the narrow-road. The goal of the experiment is to observe the effect of distance between transmitter and receiver, on the measured wireless link characteristics. The experimental setup is shown in Fig. 11. Directly measuring d' is difficult on a busy road with vehicles passing by. So we measure d''.

Keeping the transmitter fixed, we move the receiver from d''=5m to d''=40m in steps of 5m and let the receiver log for 5 minutes at each position. We do this both in free-flowing and congested conditions. The CDF of reception and RSSI, for both traffic states, are shown in Fig. 12 and 13 respectively. LOS indicates line-of-sight condition in free-flow and NLOS, non line-of-sight condition in congestion. We show the plots only for 15m, 20m, 25m and 30m to prevent the figures from getting cluttered.

As we can see, both free-flow (LOS) and congested (NLOS) traffic show almost identical PRR for $d'' < 25$ m. This is because, signal quality is quite good at small distances such that NLOS does not have much effect on the reception of packets. RSSI, which directly measures signal quality, however, shows good difference between the two traffic states even for small d'' (Fig. 13) . Thus if PRR has to be useful as a feature to detect traffic state on a narrow road, the tx-rx pair have to be placed diagonally across that road, instead of being perpendicular, to increase the effective distance between them. But such road specific sensor topology adjustments can be avoided, if RSSI alone is used as a feature.

[3]We use XBEE radios in our hardware prototype.

3.6 Classification models: summary

Thus we use FeatureClassifier algorithm with K-Means model built over 20 secs of RSSI percentiles in our *Kyun* Queue system. The deployed *Kyun* Queue system, described in Section 6, uses the training model built from the 8 hour dataset collected from the same road, with approximately 4 hours of free-flowing and 4 hours of congested data. Efficient training model building for varieties of roads and to detect drifts in model with time and environmental changes are interesting aspects of future work.

4 Design of the *Kyun* Queue system

As seen in the previous sections, one pair of tx-rx across road can infer the road occupancy level between them with significant accuracy. Next we seek to extend this pairwise sensing to build an array of sensors, which can perform co-ordinated sensing and detect length of traffic queue on a given stretch of road.

(1) Three pairs of transmitter (T_i) - receiver (R_i) are shown as example. Each pair performs sensing and computation to know the traffic condition between them. The number of pairs to be placed on a given road stretch, would depend on the worst case length of traffic queues on that road.

(2) The individual observation of each T_i-R_i pair, can be communicated to a central controller unit (C), that may reside on the traffic light. C, upon receiving the road occupancy observation values from sensors on each incoming lane, can compute the optimal green light distribution. This can handle applications local to a particular intersection. e.g. minimizing worst case waiting delays.

(3) C can communicate with a server (S), that may reside in a remote traffic control office. The server, upon receiving road occupancy information from several controller units of the city-wide road network, can implement other applications like co-ordinated signal control, bottleneck identification and congestion mapping.

4.1 Architecture

The proposed system architecture is shown in Fig. 14.

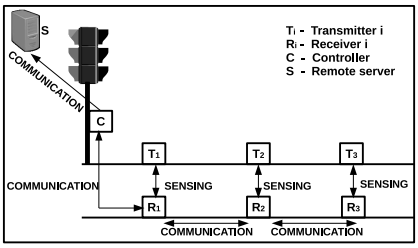

Figure 14: System Architecture

4.2 Which RF technology to use for sensing

Though we start with 802.15.4 to sense traffic as we have easy access to Telosb motes, we also experiment with 802.11 a, b and g radios and Bluetooth. The experimental setup is similar in all cases. We keep a tx-rx pair across road, at about 0.5 m from the ground and measure RSSI and packet reception rate at the rx. Power requirements for 802.11 a/b/g are higher than 802.15.4 [32]. Bluetooth shows an issue of poor range and links are hard to establish for wide roads of 25 m. Secondly, being a connection oriented protocol, it has issues of long delays in re-establishing connection following packet losses. Thus, though all the technologies are affected by congestion to some extent and can perhaps be used for traffic sensing – based on accuracy, power, range, form-factor etc., 802.15.4 seems to be a better choice than the others.

4.3 Sensing and communication conflicts

In our system, we have two types of wireless links – (1) *sensing links*, across the road, from T to R and (2) *communication links*, along the road, from one R to another R. The *sensing links* should be affected by the traffic flow on the road, so that wireless link characteristics measured on them reflect the traffic volume. This necessitates the radios on T and R to be at a low height of about 0.5 m from the ground level, such that the packets are blocked by the body of the vehicles. On the other hand, in a typical deployment scenario, the R units will be mounted on road-side lamp-posts. Inter lamp-post distance is in the order of 30 m. Even if we put our units on each lamp-post, any network fault might need units on alternate lamp-posts to communicate. Thus, we should keep provision for at least 60 m long *communication links*. These links have to be reliable as well. The question thus arises: can a single radio handle both sensing and communication links with conflicting requirements? This is verified next, through a set of experiments.

Figure 15: Obstacles on sidewalk: pedestrians and impatient motorcyclists too!

We keep a Telosb mote stationary which transmits 25 packets per second. Another mote is kept at distance x m from the transmitter; we vary x=30 m to x = 100 m, in steps of 10 m, logging number of packets received for 5 minutes at each distance. Both motes are 0.5 m above the ground. They are placed along the road, on the sidewalk of Adi Shankaracharya Marg, instead of being placed across the road like in previous experiments, since we want to evaluate the communication and not the sensing links. We do the experiment once at 6 am in the morning, when the sidewalk is empty and again at 8 pm in the evening, when the sidewalk is crowded with pedestrians (see Fig. 15) and repeat both experiments over four days.

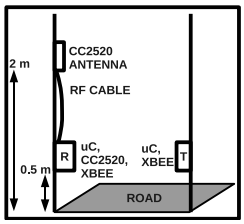

Figure 16: Sensor pair setup

The median packet reception ratio (PRR) is 60% at 60 m at 6 am. The link quality degrades at 8 pm when pedestrians on the sidewalk block the line of sight between the motes. Median PRR is at most 40% at any distance above 30 m. Such low PRR would make our communication links unreliable, and hence unsuitable for applications like adaptive traffic signal control, where timeliness is critical.

To resolve this issue, we choose to use *two* 802.15.4 radios in our receiver (R) units - one XBEE radio for sensing and a CC2520 radio for communication. The setup for a tx-rx pair is shown in Fig. 16. T is the tx unit across the road containing a microcontroller and an XBEE radio to transmit sensing packets. R is the rx unit with a microcontroller, an XBEE radio to receive sensing packets and a CC2520 radio to communicate to other R's. The CC2520 antenna is placed higher using RF cable, at about 2 m from the ground, clear of pedestrians on the footpath. Other possible design choices to handle the sensing-communication conflict, are discussed in Section 8.

4.4 Software protocol

To correlate the traffic state decisions of different R units and calculate the queue length, we have to ensure that all R's perform sensing and computation simultaneously, so that their individual measurements are co-ordinated in time. Since our architecture has a C unit, we choose to use centrally controlled measurement cycles, triggered by C, to achieve this. Thus we do not need any explicit time-synchronization mechanism among the units.

One possible way to design the software flow of our network is outlined in Fig. 20. The R units remain in receiver ready mode of CC2520 radio (C-RDY) on power up, waiting for a control message. C, on power up, asks for the current time from a server over GPRS and initializes its real time clock. C then sends a control message which each R re-

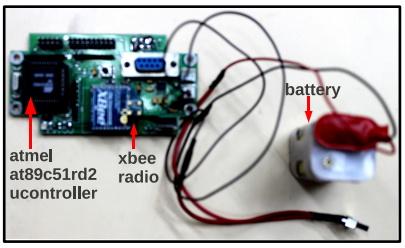

Figure 17: Transmitter (T)

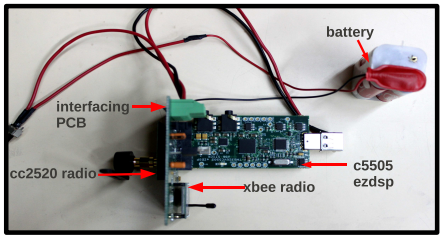

Figure 18: Receiver (R)

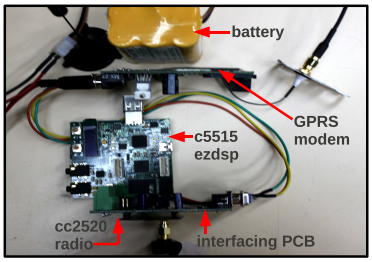

Figure 19: Controller (C)

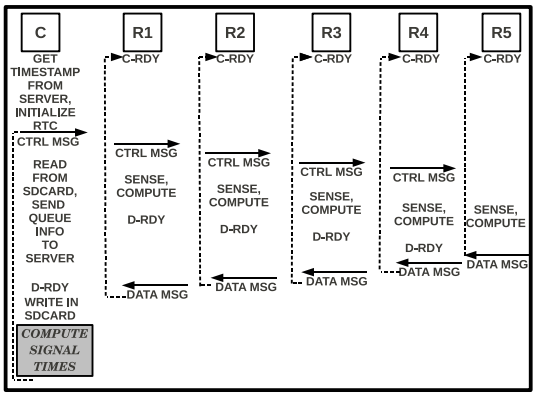

Figure 20: Software flow

ceives and transmits to the next R. After transmission, each R starts sensing the incoming packets from T on the XBEE radio and computes its binary decision of traffic state based on the algorithm described in Section 3. Once this decision is ready, each R enters receiver ready mode of CC2520 radio (D-RDY), waiting for the data message from next R. The last R creates a data message to send its decision to the previous R. Each R, upon receiving the data message from next R, appends its own decision to the message and transmits it to the previous R.

When the data message reaches C, it writes the data message, along with the time of its reception, in the Micro-SD card. This ensures retention of data and control state information, in case C reboots or server communication goes down temporarily. *C can compute the traffic signal schedule from the queue length information in the data message.*[4] All R units go back to C-RDY after transmitting data message. C appropriately sends the next control message, when it intends to start the next measurement cycle and this goes on in a loop. While the R's do sensing and computation, C reads the Micro-SD card and sends the data message of the previous cycle to the server over GPRS. The server can use this information *from different C units in the city for co-ordinated signal control* or for visualization of congestion maps.

Next we consider the MAC protocol to use in our network. TDMA needs strict time-synchronization among units. On the other hand, by design, both our control and data messages are transmitted sequentially by one R followed by the next R. Thus simple CSMA-CA can handle our MAC issues, and this is what we use in our network. To increase reliability, all messages are transmitted four times. If there is still a message loss, our design handles it by using a timeout in D-RDY state. Upon timeout, the unit goes back to C-RDY state and participates in the next measurement cycle when the next control message comes. Thus the fixed number of retransmissions allows us to achieve a good balance between resilience to stray wireless losses and implementation complexity.

C keeps track of the current measurement cycle number by generating and inserting a sequence number in the control message. If C reboots, it looks up the last sequence number from the Micro-SD card and generates the next one. If an R reboots, it simply waits in C-RDY and copies the sequence number of the first control message it gets, as the current sequence number. None of the R's can generate a sequence number. This ensures that though the sequence numbers wrap around after 0-255, there is no stale sequence number in the network. Thus units can confidently reject messages containing sequence numbers already seen or which are out of order, as retransmitted messages.

If the C and R nodes are arranged along a road, as shown in Fig. 14, this software protocol uses links between C and R_1, the first R_i along the road, and then between each R_i and R_{i+1}. Our system, using dual radio, has provision for longer communication links between R_i and R_{i+2}, which the current software does not utilize. All the messages are routed along the hardwired path of consecutive $\{R_i,R_{i+1}\}$ pairs. The longer links may be used for RSSI based self-localization, which we discuss in Section 7. Also in future, faults where a particular R_i fails or link between any consecutive $\{R_i,R_{i+1}\}$ fails, can be detected or corrected using the longer links.

5 Hardware prototypes

Based on the design choices outlined above, we have implemented three hardware prototypes for T, R and C. The T units have (1) an Atmel AT89C51RD2 microcontroller and (2) an XBEE radio on UART to transmit sensing packets. In the R units, we have used (1) a TI C5505 ezdsp stick for the computation, (2) a TI CC2520 radio connected on SPI for communication with other R units and the C unit and (3) an XBEE radio on UART for sensing. Four 1.5V batteries are

[4]The italicized parts of the software flow like traffic light schedule computation and co-ordinated traffic signal control have not been implemented. Negotiations for these are currently under progress with [9].

connected in series to provide 6V power to each circuit. The C unit has (1) a TI C5515 ezdsp stick for processing with (2) an integrated Micro-SD card for data storage. There is (3) a TI CC2520 radio on SPI to communicate with the R units and (4) a SIMCOM SIM300C GPRS modem to communicate to a server. An 11.1V, 1 Amp peak current battery is used for powering up this circuit.

The choice of the TI ezdsp sticks are for their small form factors, low cost, low power requirements and convenient pinouts of several interfaces like UART and SPI. These make hardware integration easy. The vast resource of TI chip support library functions and up to 100 MHz clock rates make programming on these platforms convenient as well. The two IEEE 802.15.4 compliant radios, needed in the R units, are chosen to be CC2520 and XBEE. This is because they use two different interfaces of SPI and UART respectively, which simplifies prototype design and implementation.

A C5505 ezdsp stick costs $50, a C5515 ezdsp stick costs $80, a GPRS modem costs $70, an XBEE radio costs $18, a CC2520 radio costs $50, an AT89C51RD2 microcontroller costs $4. With interfacing PCB's, connectors and batteries, a receiver (R) - transmitter (T) pair costs about $200 and the controller (C) costs about $250.

6 Deployment based evaluation

What is the accuracy of online traffic state classification by a single sensor pair? What is the accuracy of queue length estimates given by a network of such pairwise sensors? Are the queue estimates available in real time at the server? Can we see any pattern in the queue lengths over a day or between days? We seek to evaluate these questions through a deployment of our sensor network. We perform all our experiments on a stretch of Adi-Shankaracharya Marg road in Mumbai. This road is about 25m wide and has fair amount of traffic throughout the day as it connects two express highways.

To evaluate the accuracy of online classification, we place a T unit on a divider lamp-post and an R, perpendicularly across the road on the sidewalk lamp-post. R has a stored K-Means clustering model using nine RSSI percentile values corresponding to 10^{th}, 20^{th}, ..., 90^{th} percentile, measured over $t = 20$ secs, as features. This model is computed offline from the 8 hours (approximately 4 hours each of free-flowing and congested traffic) dataset, collected on the same road about one year back. Using the stored model, we then run the online FeatureClassifier algorithm on R, to classify chunks of 20 seconds of RSSI percentiles of packets coming from T. We experiment for 50 minutes in each of free-flow and congested traffic. 149 out of 150 data points are correctly classified as free-flow and all of 150 data points are correctly classified as congested. Thus overall accuracy of online classification on this particular set of unseen test data points is 99.67%.

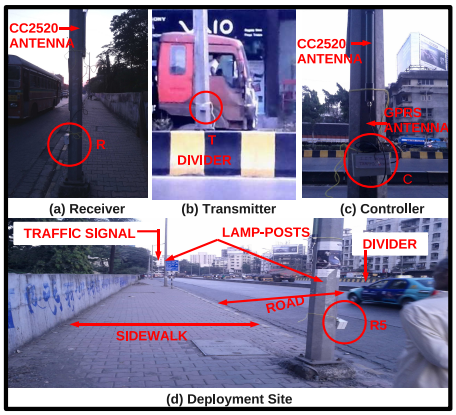

Figure 21: Deployed Units

Figure 22: View from Android Phone

Next we seek to evaluate the accuracy and timeliness of queue estimates given by our network of sensors. We deploy one C unit on a lamp-post near to the traffic signal and five T-R pairs on the next five consecutive lamp-posts. All units are packaged in ABS plastic boxes with holes to bring out the XBEE and CC2520 radio antennas. The C and the R units are clamped to lamp-posts on the sidewalk, at about 0.6m above the ground. Their CC2520 antennas, connected to the radio external ports with RF cables, are tied vertically to the lamp-posts at 2m above the ground. The C unit has an additional GPRS antenna. The T units are clamped to lamp-posts on the divider, perpendicularly opposite to the R units, such that a T-R pair face each other across the road. The deployed units and the deployment site are shown in Fig. 21.

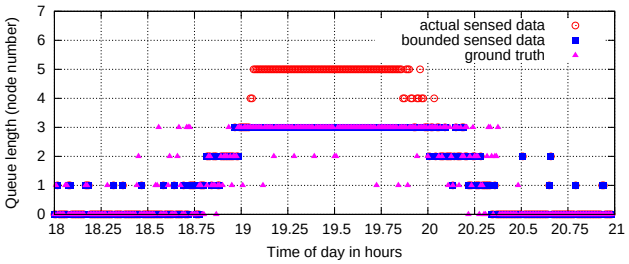

Figure 23: 6 - 9 pm, Nov 17, 2011, Saturday

We implement our designed software for this deployed network, such that C initiates a new measurement cycle through a control message every 30 secs. The R units sense for 20 seconds each and compute individual binary decisions about the traffic state. These decisions are communicated to C in a data message. C stores the message in Micro-SD card and also sends it over GPRS to a remote server. The message

Date	# of detections	% Exact matches	% Error of 1 unit (about 30m)	% Error of 2 units (about 60m)	% Error of 3 units (about 90m)
Nov 17	359	74.09	20.05 (fp=14.76, fn=5.29)	4.45 (fp=2.23, fn=2.23)	1.39 (fp=1.39, fn=0)
Nov 18	359	96.37	1.67 (fp=1.67, fn=0)	1.39 (fp=1.39, fn=0)	0.56 (fp=0.56, fn=0)
Nov 19	353	90.93	7.64 (fp=3.39, fn=4.24)	1.97 (fp=0.28, fn=1.13)	0 (fp=0, fn=0)
Overall	1071	87.11	9.8 (fp=6.62, fn=3.17)	2.4 (fp=1.3, fn=1.12)	0.65 (fp=0.65, fn=0)

Table 2: Accuracy and error breakups of deployment results

is in the form of an array of 5 binary values, each signifying decision by an R, in increasing order of the lamp-posts from the C unit. The server logs these updates coming every 30 seconds and computes the queue length as the unit number of the last R reporting congested state. Thus our measured queue lengths can take 6 discrete values: 0, where all R's report free-flow, 1 when only R1 reports congestion while the others report free-flow, 2 when R1 and R2 report congestion while others report free-flow, 3, 4 and finally 5, when all R's report congestion. We run this deployed network on Nov 17, Thursday, Nov 18, Friday and Nov 19, Saturday, 2011, for 3 hours everyday, between 6-9 pm.

To know the accuracy of our measurements, we use an image-based manual verification scheme. We run an Android application on a Samsung Google Nexus phone to capture an image every 30 second and store it. The phone is placed on the roof of a four storeyed building by the roadside. The phone can cover T-R pairs 1, 2 and 3. We tried different apartment buildings by the roadside and different orientations and zoom-levels on the phone, but this was the maximum number of sensors that we could cover. The view from the phone is shown in Fig. 22. In the figure, C is further to the left of sensor pair 1 and T-R pairs 4, 5 are further to the right of sensor pair 3. The images for the three days can be viewed at [31]. One person observes the images offline and estimates the queue lengths manually. In case this observer finds it difficult to estimate the length from the image, a second observer is consulted and the queue length is ascertained by their mutual consent.

The accuracy and break up of errors of our deployment results are summarized in Table. 2. Error of 1 unit indicates that our queue estimate and the ground truth differ by 1. This in turn can be a case of false positive (fp) or false negative (fn). The false positives and false negatives are determined in the following way. The viewer of the image determines the current queue length by seeing the image. This is considered as the ground truth. If the queue length reported by our sensor system is more than the ground truth queue length, that is considered as false positive, as we are overestimating the queue. Similarly, if the reported queue length is less than the ground truth queue length, that is considered as false negative, as we are underestimating the queue. Error of up to 3 units can occur, as images cover units 0-3.

Fig. 23, Fig. 24 and Fig. 25 show the length of the queue as measured by our system on the three consecutive days respectively. To aid visual comparison with ground truth which covers up to unit 3, we plot both the actual queue values reported by our system, termed as *actual sensed data* in the figures, and the min(3,actual sensed data), termed as *bounded sensed data* in the figures.

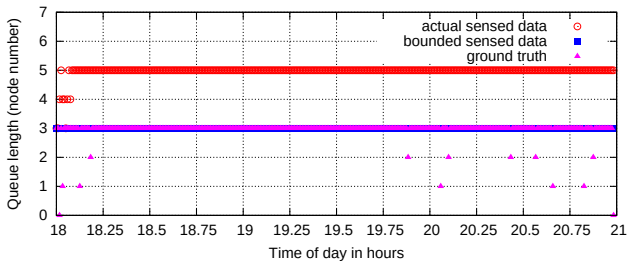

Figure 24: 6 - 9 pm, Nov 18, 2011, Saturday

As we can see from Table 2, the accuracy is upto 96% on Nov 18 and Nov 19. Of the three days, Nov 17 has minimum accuracy: 74%. But as seen from Fig. 23, queue buildup and clearing was very rapid on that day, increasing the challenge of deciding queue length by manual observation. So our low accuracy is likely a combined effect of our errors and errors of manual ground truth estimation. For example, the instant the image is taken, the queue might have cleared but it might have been present for most of the 20 seconds of sensing, leading to a false positive. A case of false negative would occur if a queue builds up the instant the image is taken, while most of the sensing time, traffic is free-flowing. A better way of ground truth estimation would be to take continuous video, instead of an instant image, and we accept this to be a limitation of this work. False negatives are rare for the error values of 2 and 3, as instant growth and reduction of long queues is non intuitive. But even on Nov 17, almost all the errors are only of one unit and higher errors of 2-3 units, given in the last two columns of Table 2, are very low. The above results indicate that, the *Kyun* Queue system is accurate in estimating traffic queue lengths, and shows good promise for use in applications such as automated traffic signal control.

An interesting point to note from the figures is: queue length can be fairly variable (1) over three hours on a single day, as seen in Fig. 23, (2) between two week days at the

same time of the day, as seen between Fig. 23 and Fig. 24 and (3) between weekdays and weekend, as seen between Fig. 23, Fig. 24 and Fig. 25. This variability further motivates the usefulness of our system in building applications like dynamic traffic light control, to make traffic management more reactive to current traffic status.

An interesting aspect of evaluating our system would be to compare it with other existing sensing mechanisms like images and IR. However, direct comparison with the current implementations would be biased towards our system, because the other implementations are meant to be used in orderly traffic. We however, have specifically built our system to handle chaotic traffic. Re-implementing all other techniques for chaotic traffic ourselves, on the other hand, is definitely beyond the scope of one technical paper.

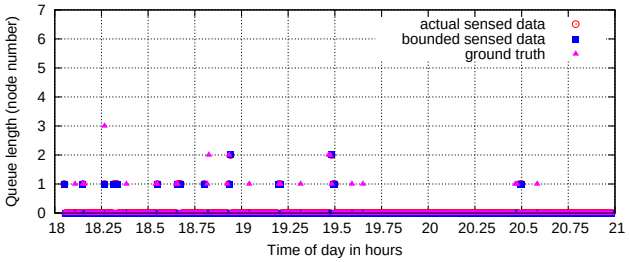

Figure 25: 6 - 9 pm, Nov 19, 2011, Saturday

7 Self localization

We have thus far evaluated the effectiveness of the *Kyun* Queue system. However, to calculate the traffic queue length from individual R decisions, the ordering of the R units with respect to C has to be known, to correlate which R decision represents which part of the road. In this section, we explore whether we can have a built-in localization module, to configure this sensor ordering without manual intervention. This will be useful in real-life deployment scenarios, where hundreds of such networks will be needed to be set up, one network for every road intersection in a city.

We can have statically addressed R's and deploy them carefully in a pre-defined order along the road. Then each R unit will have to be programmed with a different software containing a different node-id, and deployment also has to be careful to obey the ordering.[5] A second option is to have a keypad attached to every R. All R's are programmed with the same software and deployed in any order. After deployment, keys are pressed at each R, to enter the addresses in increasing order along the road. A third option is to use a GPS-enabled smartphone during deployment and record the GPS co-ordinates corresponding to each sensor ID. All these need technical proficiency of the deployers, an unnecessary additional requirement.

A fourth option is to use RSSI ranging, as both R and the C have CC2520 radios with antenna anyway. The R's are at gradually increasing distance from C, which might reflect in the RSSI of packets sent by C and received at the different R's. We experimentally verify the fourth option, as it promises minimum cost and deployment overhead.

Self localization of sensor nodes is one of the most researched areas in the field of sensor networks. But strangely, none of the proposed solutions are used in actual deployments to the best of our knowledge. Most deployed networks [33, 34], have statically addressed nodes, carefully placed in pre-decided locations. Some deployments handle applications like bridge monitoring [35], flood [36] or forest fire [37] detection, each of which compulsorily needs the location of the sensed data. But the localization procedures are not specified in these papers. A few like [38] have used GPS to determine sensor location. [39] recently attempted to approximately localize a very large deployment of sensor nodes using RSSI ranging.

The main challenge in using RSSI is that, it is not a monotonically decreasing function of distance. The theoretical two-ray ground refection model has found strong empirical evidence in [35], which shows that RSSI oscillates with distance due to multipath interference. When we plot the two-ray model equations in Matlab, lower antenna heights seem to give less oscillations (Fig. 26). But our deployment location being road-side lamp-posts on the sidewalks, very low antenna height will affect communication among R's due to obstruction by pedestrians, as experimentally shown in Section 4. This will defeat our purpose of having a dual radio platform. A minimum height of 2 m seems to be necessary, considering average human height to be less than that.

Secondly from Matlab plots, vertical polarization seems to give far less oscillations than horizontal polarization (Fig. 27). We seek to verify this experimentally, by placing a CC2520 transmitter on a lamp-post and moving a CC2520 receiver gradually away from one lamp-post to the next, with antenna at 2 m height for both radios. The CDF's of RSSI are plotted in Fig. 28. The two antennas are first placed horizontally, perpendicular to the lamp-post, pointing towards the sidewalk, denoted by *i*-h in Fig. 28, $2 <= i <= 6$ being the five consecutive lamp-posts where the receiver radio is kept. Another set of readings are taken by keeping the antennas vertical, parallel to the lamp-posts, denoted by *i*-v in Fig. 28. According to the order of the RSSI CDF's, the lamp-posts would be ordered as {node 2, node 3, node 5, node 6, node 4} for horizontal and {node 2, node 3, node 6, node 5, node 4} for vertical orientations of the antenna. The re-ordering of lamp-posts occur independent of antenna orientation and hence polarization. A secondary thing to note is, vertical orientation gives worse RSSI values than horizontal; this is likely due to fading effect of the metal body of the lamp-post with which the vertical antenna is parallel.

[40] proposed that by using different 802.15.4 channels, the average RSSI over the channels would have less oscillations. But our Matlab plots show little difference for the 16 channels, four of which are plotted in Fig. 29 as example. Thus none of the parameters like antenna height, polarization or channel, seem to be useful to remove the RSSI oscillations.

We can see the effect of these oscillations in Fig. 30 and Fig. 31. Fig. 30 shows CDF of RSSI of packets received at a

[5] This is what we do in our current deployment.

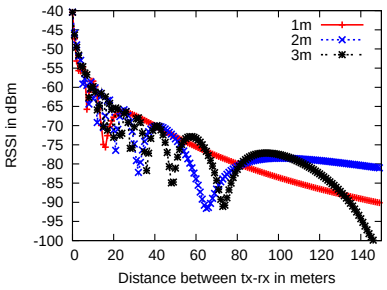

Figure 26: Height Effect (simulated)

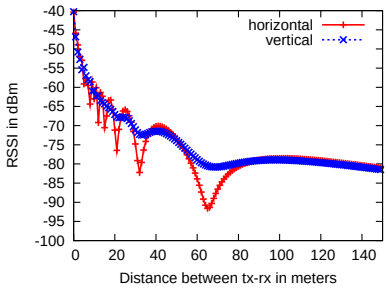

Figure 27: Polarization (simulated)

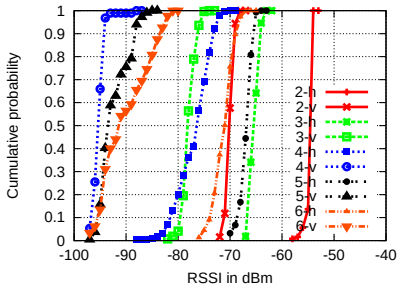

Figure 28: Polarization (empirical)

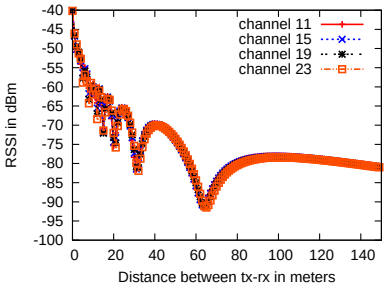

Figure 29: Channel Effect (simulated)

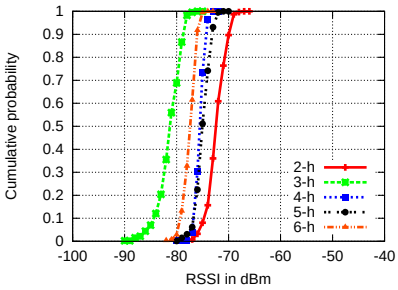

Figure 30: Sender at lamp-post A

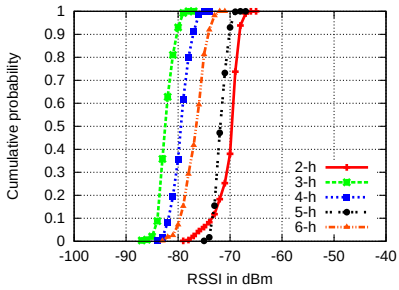

Figure 31: Sender at lamp-post B

receiver CC2520, which is moved from one lamp-post to the next, keeping a transmitter CC2520 fixed at one lamp-post. This is repeated by keeping the transmitter in another lamp-post (Fig. 31). According to the order of RSSI CDF's, we would order the lamp-posts as {node 2, node 5, node 4, node 6, node 3} and {node 2, node 5, node 6, node 4, node 3}. The transmitter-receiver distances for the five positions are 31, 70, 95, 116 and 144 meters and 39, 64, 85, 113, 140 meters respectively. Distances 31 and 39 m have the best RSSI, being at the initial steep drop region of the curve (Fig. 29). 70 and 64 m have the worst RSSI, being at the lowest dip region of the curve. Remaining distances have comparable RSSI, lying in the final flat part of the curve. Thus the RSSI closely conforms to the 2-ray model (Fig. 29). Thus an exact match between lamp-post order and RSSI order seems difficult, if not impossible, even within a single road stretch! The RSSI oscillations are compounded by the fact that inter lamp-post distances vary as well. We measured inter lamp-post distances of 20, 31, 39, 25, 21, 28, 27 and 29 meters in our deployment location.

On the brighter side, our results above indicate that the following rule of thumb may work: transmit packets from C and measure RSSI at all R's. The R recording the best RSSI is marked as the first node. Now this first node transmits packets and all other R's measure RSSI. The R recording the best RSSI is marked as the second node. This process continues iteratively, until all the R's are marked. Localization is a one time process, at the time of initial deployment, so the duration of the process is not a concern.

Moreover, since the growth of traffic queue lengths generally follow a typical pattern of growing outwards from the signal, the sensor pairs near the signal will typically detect congestion earlier than the pairs away from it. Thus even if one or more sensors are wrongly localized, comparison of the reported congestion values by all pairs over some days, should identify the localization inversion.

With the multitude of literature on localization using RSSI [26, 27, 28, 29], one might think that localizing a linear array of sensors based on signal strength will be trivial. However, as we empirically show, the process is not straightforward and there are several challenges that come up. Finally we propose a simple iterative mechanism that might work in practice.

8 Discussion and future work

In this section we discuss some aspects relevant to our system and outline a few areas of future work.

Interference issues - If all the T units transmit on the same channel, hidden node problems can occur in our network. For example, as shown in Fig. 32(i)(a), T1 and T3 might transmit simultaneously, though XBEE radios use CSMA/CA, as they might not hear each other. Collision might happen at R1, if it can hear both T1 and T3, which is possible because of wireless link asymmetries. 802.15.4 has 16 orthogonal channels. Thus each T-R pair can potentially use a different channel to avoid intra-network interference. But inter-network interference can still occur, because of Wi-Fi interference from nearby 802.11 access points.

Fortunately, interference does not affect RSSI of received

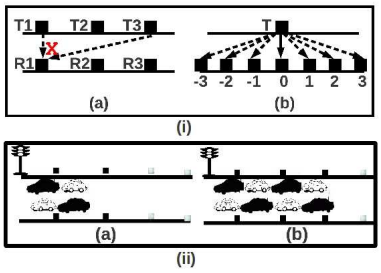

Figure 32: Interference & duty-cycling

packets [41, 42]. If Clear Channel Assessment (CCA) is disabled at tx, it stops holding back packets in presence of interference [43]. If Cyclic Redundancy Check (CRC) is disabled at rx, all packets, even with bits corrupted by interference would be passed to the application layer. Since we use only RSSI for sensing, the actual packet content is irrelevant to us. Disabling CCA and CRC, increases number of received packets in presence of interference, using RSSI of which, our technique would work fine. Only the communication of sensed data needs an interference free channel, as there the actual packet contents are important. But for this we can use 802.15.4 channel 26, which lies outside the 802.11 spectrum. Our preliminary experiments disabling CCA and CRC show the usability of all channels for sensing, even in presence of heavy 802.11 interference. This will be explored further in future.

Power consumption - Battery life saving is not a very crucial requirement for our system, as our units are deployed on street lamp-posts with constant power supply from grid lines. Still, to get a power budget for the batteries used in our prototype deployment, we explore our system power consumption. The T units perform 20 XBEE tx operations per second. The R units, in every measurement cycle spanning 30 secs, receive at most 400 XBEE packets during sensing, perform one classification operation and receive and transmit at most eight CC2520 messages. The C unit, in every measurement cycle of 30 secs, performs one GPRS communication, two SD card operations and receive and transmit at most eight CC2520 messages. The power consumed for each individual operation is given in Table 3.

Function	mW	Function	mW
C5505 operations	213	GPRS	2016
CC2520 tx (0 dBm)	167	CC2520 rx (0dBm)	610
XBEE tx (0 dBm)	390	XBEE rx (0 dBm)	540

Table 3: Function Specific Power Consumption

A power optimization approach, that we may consider in future, is as follows. It might be unnecessary to keep all T-R pairs functioning at all times. For e.g., in our Nov 19 deployment (Fig. 25), the first two sensor pairs would have been enough to detect queues. A simple duty-cycling mechanism can be as follows. The pair nearest to the signal remains awake at all times. If this pair sees congestion for more than a threshold number of cycles, the next pair wakes up. This continues as the queue grows and the whole network gradually comes up. This is illustrated in Fig. 32(ii), where dark rectangles denote nodes which are awake and the others denote sleeping nodes. Some power can also be saved by suppressing updates that can be inferred by correlating other updates [44].

Classification schemes and model training - Semi-supervized training of the classification models for different kinds of roads will be studied, to balance between overhead of manual labeling in supervized learning and noise in unsupervized methods. Secondly, whether classification model built for one road works on other roads, will be explored. Specifically, we will try to identify characteristics like road width and vehicle types, that affect model parameters, and subsequently see if roads similar in those characteristics can use the same model with sufficient accuracy. Thirdly, instead of binary classification of traffic states by each sensor pair, whether multilevel classification of (a) empty road, (b) fast traffic, (c) slow traffic and (d) standing traffic is achievable with a single sensor pair, using an enhanced set of features and algorithms, is another area to explore. Fourthly, classification model drift with time due to effect of weather and other environmental factors would be studied over long-term data, with special focus on automatic drift detection applying concept drift theory of machine learning. Also the minimum amount of training data required to achieve a given classification accuracy will be interesting to quantify.

Other design choices - Finally, it will be interesting to try different network topologies, instead of the linear array, to reduce hardware overhead. An example star-topology is shown in Fig. 32(i)(b). A second design choice to explore, would be to replace the dual radio solution with single radio, which has two antenna ports [45]. We might attach an external antenna to one port and keep the other empty and can select the former for communication and latter for sensing, dynamically in software. A third design choice to explore is the use of directional antennas. Such antennas have fairly low costs and they have been shown to improve the stability of wireless links [46]. This may render the RSSI measurements to be less susceptible to oscillations, possibly further improving the detection accuracy. Further, customized platform design and cheaper radios and microcontrollers will be considered to bring down overall system costs.

9 Conclusions

Traffic monitoring in developing regions poses a set of challenges, not met fully by existing systems. In this paper, we have designed and implemented a new sensing system to detect road occupancy based on RF link quality degradation. We have also designed and implemented a sensor network to distributedly decide traffic queue length in real time. Our system, deployed on a Mumbai road, achieves upto 96% accuracy in queue length estimation. Thus it shows good promise to be immediately useful for a variety of applications in real situations.

10 References

[1] J. Yoon, B. Noble, and M. Liu. Surface street traffic estimation. In *The 5th International Conference on Mobile Systems, Applications, and Services (MobiSys '07).*

[2] E.Koukoumidis, L.Peh, and M.Martonosi. Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory. In *The 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*.

[3] R. Sen, P. Siriah, and B. Raman. Roadsoundsense: Acoustic sensing based road congestion monitoring in developing regions. In *The 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '11)*.

[4] www.pluggd.in/indian-traffic-entrepreneurial-lessons-297/.

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

[6] http://www.scats.com.au/index.html.

[7] http://paleale.eecs.berkeley.edu/~varaiya/transp.html.

[8] http://www.visioway.com, http://www.traficon.com.

[9] http://www.mapunity.in/.

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

[11] I. Magrini G. Manes A. Manes B. Barbagli, L. Bencini. An end to end wsn based system for real-time traffic monitoring. In *The 8th European Conference on Wireless Sensor Networks (EWSN '11)*.

[12] R. Sen, B. Raman, and P. Sharma. Horn-ok-please. In *The 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*.

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

[14] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*.

[15] R.Balan, N.Khoa, and J.Lingxiao. Real-time trip information service for a large taxi fleet. In *The 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*.

[16] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *The 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*.

[17] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 2003.

[18] A. Quinn and R. Nakibuule. Traffic flow monitoring in crowded cities. In *AAAI Spring Symposium on Artificial Intelligence for Development*, 2010.

[19] V. Jain, A. Sharma, and L. Subramanian. Road traffic congestion in the developing world. In *The 2nd Annual Symposium on Computing for Development (DEV '12)*.

[20] Skordylis A. and Trigoni N. Efficient data propagation in traffic-monitoring vehicular networks. *IEEE Transactions on ITS, volume 12*, 2011.

[21] Leontiadis I., Marfia G., Mack D., Pau G., Mascolo C., and Gerla M. On the effectiveness of an opportunistic traffic management system for vehicular networks. *IEEE Transactions on ITS, volume 12, issue 4*, 2011.

[22] Skordylis A. and Trigoni N. Delay-bounded routing in vehicular ad-hoc networks. In *the 9th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*.

[23] Yuan J., Zheng Y., Xie X, and Sun G. T-drive: Enhancing driving directions with taxi drivers intelligence. In *Transactions on Knowledge and Data Engineering (TKDE '12)*.

[24] H. K. Le, J. Pasternack, H. Ahmadi, M. Gupta, Y. Sun, T. Abdelzaher, J. Han, D. Roth, B. K. Szymanski, and S. Adali. Apollo: Towards factfinding in participatory sensing. In *The 10th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN '11)*.

[25] V. Shrivastava, D. Agrawal, A. Mishra, and S. Banerjee. Understanding the limitations of transmit power control for indoor wlans. In *The 7th Internet Measurement Conference (IMC '07)*.

[26] Moustafa Y. and Agrawala A. The horus wlan location determination system. In *The 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*.

[27] Chintalapudi K., Iyer A. P., and Padmanabhan V. N. Indoor localization without the pain. In *The 16th Annual International Conference on Mobile Computing and Networking (MobiCom '10)*.

[28] Patwari N. and Wilson J. Rf sensor networks for device-free localization: Measurements, models, and algorithms. In *Proceedings of the IEEE, volume 98, number 11*, 2010.

[29] Xu C., Firner B., Zhang Y., Howard R., and J. Li. Statistical learning strategies for rf-based indoor device-free passive localization. In *The 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*.

[30] Mottola L., Picco G. P., Ceriotti M., Gună Ş., and Murphy A. L. Not all wireless sensor networks are created equal: A comparative study on tunnels. In *ACM Transactions on Sensor Networks (TOSN '10)*.

[31] Anonymous picasa album containing ground truth images and sample videos from deployment location. In *https://picasaweb.google.com/108285005574366399467*.

[32] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni. Lit mac: Addressing the challenges of effective voice communication in a low cost, low power wireless mesh network. In *The 1st Annual Symposium on Computing for Development (DEV '10)*.

[33] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *The 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*.

[34] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*.

[35] K. Chebrolu, B. Raman, N. Mishra, P. Valiveti, and R. Kumar. Brimon: a sensor network system for railway bridge monitoring. In *The 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*.

[36] E. Basha, S. Ravela, and D. Rus. Model-based monitoring for early warning flood detection. In *The 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*.

[37] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *The 4th International Conference on Mobile Systems, Applications, and Services (MobiSys '06)*.

[38] W. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *The 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*.

[39] W. Xi, Y. He, Y. Liu, J. Zhao, L. Mo, Z. Yang, J. Wang, and X. Li. Locating sensors in the wild: pursuit of ranging quality. In *The 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*.

[40] A. Bardella, N. Bui, A. Zanella, M. Zorzi, P. Marron, T. Voigt, P. Corke, and L. Mottola. An experimental study on ieee 802.15.4 multichannel transmission to improve rssibased service performance. In *The 5th Workshop on Real-World Wireless Sensor Networks (REALWSN '10)*.

[41] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee. Diagnosing wireless packet losses in 802.11: Separating collision from weak signal. In *The 27th IEEE International Conference on Computer Communications (INFOCOM '08)*.

[42] J. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *The 6th European Conference on Wireless Sensor Networks (EWSN '09)*.

[43] C. Mike Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *The 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*.

[44] Guitton A., Skordylis A., and Trigoni N. Utilizing correlations to compress time-series in traffic monitoring sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC '07)*.

[45] http://www.jennic.com/products/modules/jn5148_modules, http://www.jennic.com/products/modules/jn5139_modules.

[46] Ostrom E., Mottola L., and Voigt T. Evaluation of an electronically switched directional antenna for real-world low-power wireless networks. In *The 5th Workshop on Real-World Wireless Sensor Networks (REALWSN '10)*.