# QueueTrak: Automated Line Length Detection using a Wireless Sensor Network

Jared Alexander, Matthew Bocknek, and Kamin Whitehouse

## ABSTRACT

We describe QueueTrak, an automated wireless sensor system designed to detect and report the length of lines in retail environments. QueueTrak has two components. First, it uses a series of custom active infrared sensors to detect the length of a line in a store or restaurant. Second, a web interface publishes the data, at the discretion of the shopkeeper, for use by customers, city planners, and other shopkeepers. We empirically evaluate the system design and find that it can accurately estimate line length despite noise caused by variations in natural sunlight or moving people and furniture in the store.

## Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems

## General Terms

Design, Measurement

## Keywords

Wireless Sensor Network, Ubiquitous Computing, Information System, Detection

## 1. INTRODUCTION

Long lines in stores and restaurants are a problem for owners and customers alike. Although a small number of venues such as night clubs and restaurants use long lines to advertise their popularity, customers do not like to waste time waiting in lines and, in general, make fewer purchases at locations with long lines. People often forgo purchases in a retail store or choose to eat at a different restaurant in order to avoid long lines. People sometimes avoid a venue altogether if it is known to occasionally have very long waiting times, particularly when under a time constraint such as a lunch break during the work week.

In this paper, we present the design, implementation, and evaluation of *QueueTrak*, a sensor network that measures the length of lines at the door or cash register. It then uses wireless networking to update estimates of line length on a Web server in real time. If deployed widely, this system would allow customers to make informed decisions about where to shop or eat based on waiting times. For example, we envision information from multiple QueueTrak sensors in a town being integrated in a smart phone application that lists a person's favorite lunch venues in the order of increasing waiting times.

QueueTrak also aims to deliver numerous benefits to retail owners and management. For example, long-term data analysis could help optimize staff scheduling to better manage peak seasons, days, or hours, and this could in turn lead to faster, more efficient service and increased revenue. Line analysis could also reveal inefficiencies in a retail system. Finally, by publishing the QueueTrak data, a store or restaurant can attract customers when lines are short and can avoid making customers unhappy when lines are long.

One goal of QueueTrak is to be a *self-financing* sensor infrastructure that has the potential for widespread deployment because it provides a financial benefit to the store owner. As such, the cost of materials and installation was a key priority. Unlike existing camera-based systems that cost upwards of $700 and require power and data lines to be run through the store, the QueueTrak prototype costs only $90 in hardware, and this price is expected to substantially decrease with larger volume production. The system is also battery operated and uses wireless communication, so it does not incur wiring costs, and installation is as simple as using double-sided tape. A simple daisy chain design allows new sensors to be added by non-expert users such as shopkeepers. Finally, QueueTrak monitors line length in a completely unobtrusive way without introducing inconvenient ticketing systems, surveillance cameras, or other equipment that can change the mood or ambiance of a store or restaurant.

Once the QueueTrak system is installed and the data is published on the Web, the data could also provide many benefits to users besides the shopkeeper and customers. For example, city planners and anthropologists could aggregate the information from multiple stores to perform longitudinal studies that analyze the effect of weather, festivals, concerts, and other events on local business. In fact, one motivation of the QueueTrak system is to provide empirical data for an ongoing debate in Charlottesville, Virginia about the effect on business of vehicular traffic crossing the downtown pedestrian areas. Real estate experts could use the information to value a property. Even neighboring shops and restaurants could use long-term trends in the data to gauge their own performance in serving customers, generating competition among shops to reduce line lengths. As such, QueueTrak provides a valuable case study through which we can analyze the fundamental issues surrounding public sharing of private sensor data on the Web, such as incentives, aggregation, and privacy. A long-term goal of QueueTrak is to identify those aspects of a sensor system that align the interests of sensor owners with those of others that could benefit from the data, in order to produce an ecosystem for wide-spread data sharing on the Web.

Manuscript received May 28, 2010. This work was supported by the Computer Science Department of the University of Virginia.

J. Alexander is a Computer Engineering graduate of the University of Virginia, Charlottesville, VA 22904 USA (jaredalexander@alumni.virginia.edu).

M. Bocknek is an Electrical Engineering graduate of the University of Virginia, Charlottesville, VA 22904 USA (matthew.bocknek@alumni.virginia.edu).

K. Whitehouse is a Computer Science professor at the University of Virginia, Charlottesville, VA 22904 USA (whitehouse@virginia.edu).

## 2. RELATED WORK

Several commercial systems provide information about the number of customers and their activities in a store. For example, some sensor deployments count the number of people that pass through a particular location. These systems are usually used to count people entering a store, but could also be used to count people entering and leaving a line, and therefore to infer line length [1,2]. NICE Systems offers an alert system when a line or queue is overloaded by performing real-time analysis of a video feed of the end of the line [3]. However, this system cannot provide line length because it only monitors when the line is too long. The Department of Motor Vehicles (DMV) and other places that issue tickets to customers before they are served are able to estimate both the size of the line and the average wait times, and the DMV also publishes this information on the Web in almost real-time. However, none of these systems simultaneously address all of the goals of QueueTrak: to provide a cheap and simple system that is easy to install, unobtrusive for the customers, and publishes real-time estimates of line length on the Web.

Mechanisms for sharing sensor data are sometimes called the *World Wide Sensor Web*, and have been proposed in many forms. IrisNet stores sensor data in a distributed database which is dynamically partitioned for geographic indexing [4]. Other proposals push data through an aggregation network [5,6], and the SensorMap project at Microsoft Research uses a database-oriented approach but generates the queries from a graphical map interface [7]. In this system, we propose a very different approach in which we publish sensor data in a completely unstructured fashion by simply publishing the data to a Web page. This approach follows the design philosophy of our previous work on MetroNet, which senses people in front of a store [8].

## 3. SYSTEM DESIGN

Figure 1 depicts the use case scenario for which QueueTrak was designed. Customers line up along the railings before approaching the cash register and placing an order. Sensors would be placed along the length of the line, as indicated by the red arrows. As the line gets longer, more sensors detect people.

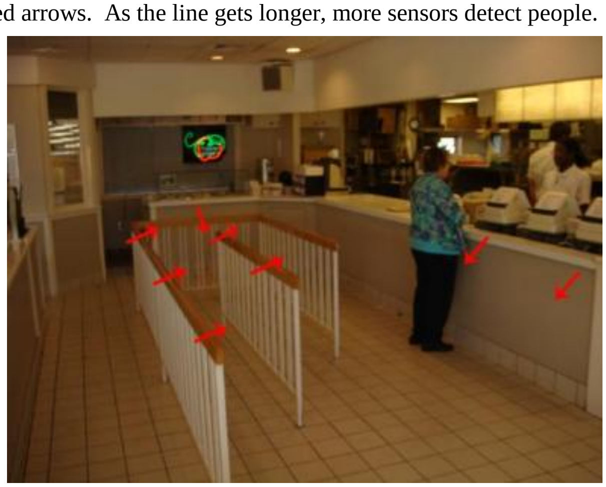

**Figure 1: Target Use Case Scenario**

The design of QueueTrak consists of three main components, as illustrated in Figure 2: (a) a series of coupled infrared sensors and actuators (b) a gateway system to collect the data, consisting of a wireless transmitter, wireless receiver, and a client machine in the store that is connected to the internet to collect the data, and (c) a backend Web server that provides data to users. Our prototype supports 16 sensors, although it can be adapted to support as many nodes as the retailer needs to cover the entire line length. The first sensor in the string is designated the master node because it controls the rest of the sensors and collects their data. The following three sections describe each component of the system design in more detail.

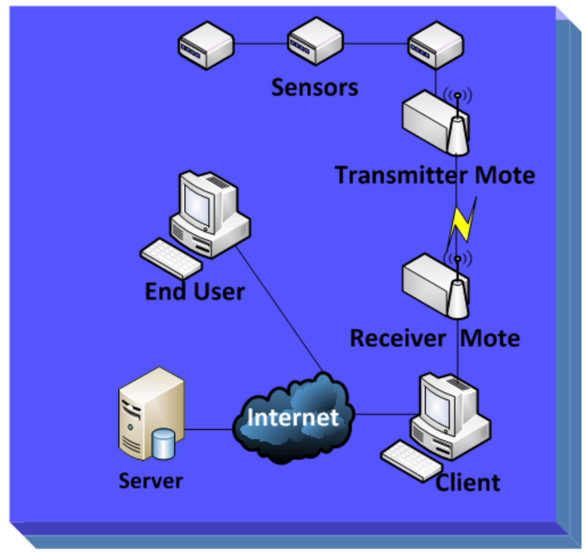


**Figure 2: System Topology**

### 3.1 Sensors

#### 3.1.1 *Initial Designs*

Since motes were chosen as the devices to communicate wirelessly, the sensors had to be designed with this interface in mind. Our first ideas were to utilize the mote's analog-to-digital-converter (ADC) in order to design purely analog sensors. The initial sensors we designed were active infrared. Each sensor had an infrared emitter and phototransistor. The infrared light bouncing off a near bystander led to a larger current through the phototransistor. This additional current flowing through the resistor created a voltage drop read by the mote's ADC and this signaled the presence of a person. However, this topology did not work as the device registered all sources of infrared light such as the sun and changing ambient light. These changes in light railed the voltage output and detection became impossible. To solve the problem of phototransistor output railing, a complex gain controlling circuit was constructed. The end result was a sensor that was viable over a greater range of ambient lighting conditions. This circuit contained an op-amp, digital potentiometer, and microcontroller. The microcontroller ran code that sampled the analog value of the op-amp output. Depending on the value obtained, it changed the value of the digital potentiometer. This digital resistor was hooked up so that it formed a feedback path on the op-amp. This circuit automatically changed its own gain, either increasing or decreasing it to maintain optimal sensitivity.

### *3.1.2 Final Design*

The final sensor design consists of microcontrollers connected via a single serial bus. Each sensor has an infrared transmitter modulated at 38kHz and an infrared receiver tuned to the 38kHz frequency. Signal detection by the receiver indicates that an object is immediately in front of the sensor and is reflecting the signal. Modulation of the infrared signal drastically reduces the amount of noise and false positive readings due to natural infrared variation in a room throughout the course of the day, as cloud cover changes, or as people and objects cover or uncover a window. Gain was adjusted by changing the brightness of the emitter using a resistor. The gain control is tuned to maximize sensitivity and increase the range of the sensor while minimizing the number of false positive detections. A schematic of the sensor is shown in Figure 3.

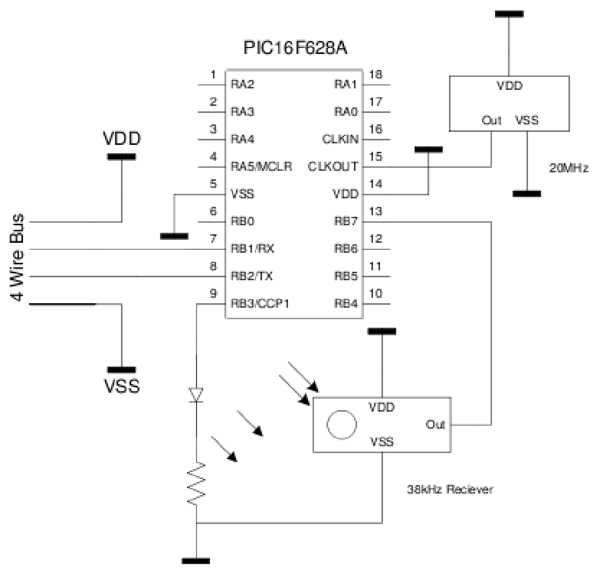


**Figure 3: Sensor Schematic**

Each coupled infrared sensor is equipped with an independent PIC microcontroller that controls the transmitter and reads from the receiver. Bus arbitration is performed by only initiating communication through the first microcontroller, which is designated the master, and all other sensors are slaves. This first sensor node in the chain is connected directly to a wireless mote. To communicate, the master node queries each sensor and waits for its reply. When a slave sensor receives a query from the master, it outputs a 38kHz pulse via pulse-width modulation for 250ms and simultaneously monitors the receiver. If the amount of detections per 250ms is greater than half, it responds to the master that the module has detected the presence of a person. This algorithm further reduces false positives by negating the effect of stray reflections. Because all microcontrollers are connected by the same serial bus, new sensors can be added in a simple daisy chain fashion, allowing even novice users to install new sensors. Each sensor has a four bit wide DIP switch that sets the node's address. If more nodes are needed, a larger address space can be created. The sensors are enclosed in discrete plastic enclosures, as shown in Figure 4.

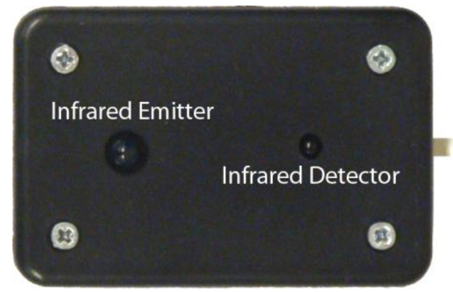


**Figure 4: Discrete Packaging of a Sensor**

## 3.2 Gateway Protocol

A Crossbow Telos Revision B mote was used as the master node in our prototype, and another mote was attached to a client computer that was connected to the Internet. The master node reports a 16-bit value for each sensor, which represents the measurement from sixteen samples taken 250 ms apart for that sensor. An example of the packet format is shown in Figure 5. Bits [0-71) store the header of the data packet, bits [71-103) store the mote ID and sequence number, and the rest of the bits store the sensor readings. In this example, sensor 1's value is DB FF. This hexadecimal translated to binary is 1101101111111111. This bit string holds four seconds of sensor data, and each one of these bits represents the presence or absence of a person in a 250 ms timeslot. If the majority of the bits are ones (as in this example), the data is interpreted as a person detected for that four second interval.

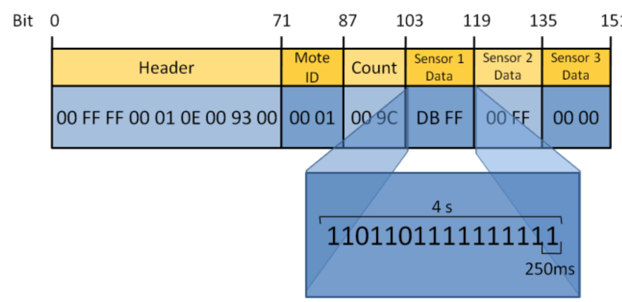


**Figure 5: Data Packet Format**

## 3.3 Server and Web Application

The server is built with an Apache Web Server, a MySQL Server, and a Java program that receives the data and inserts it into a database. The web application for QueueTrak is written in a combination of HTML, PHP, and JavaScript. The server URL serves two demographics: retailer management and customers. The retailer side is private and requires a login with a user name and password to authenticate stores. The customer side is public and requires no authentication.

### *3.3.1 Shopkeeper Interface*

The management side allows a user to view graphs of line length data over time. For our prototype, line lengths are defined as "none", "short", "medium", and "long". A user can either choose a single date to view data or a range of dates. In both cases, he or she can further specify a time range. The single date view displays a scatter plot of line lengths throughout the time range, as shown in Figure 6. The date range view displays a qualitative average of line length throughout the time range using a bar graph.

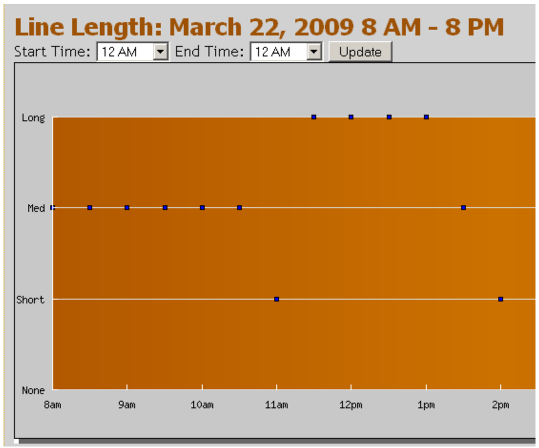


**Figure 6: Private Interface Example**

### *3.3.2 Public Interface*

The public interface provides a simple visualization of the current line length for the store. The data can be represented in many ways, including an RSS feed and HTML text. We implemented an animation that shows the current line length in real-time. Asynchronous JavaScript and XML (AJAX) was used to eliminate the need for users to refresh the page.

## 4. EVALUATION

We evaluated the QueueTrak sensor system in terms of its ability to respond to dynamic conditions using automatic gain control, as well as the accuracy with which it could detect people on line. Figure 7 illustrates the effect of automatic gain control of our early prototype. At point A, an object was placed in front of the sensor at 10 cm and then removed. Likewise, at point B, an object was placed at 20 cm and then removed. This illustrates proper sensor behavior under normal light conditions. Ambient light was increased at point C by opening a window blind, and the blue line shows the drop in output from automatic gain control, but the orange line does not drop. All of the following data was collected with increased ambient light. At point D, an object was placed at 10 cm and removed. The orange line shows the voltage value railing while the blue line with gain calibration successfully avoids the railing problem. Point E once again illustrates the output with and without automatic gain control.

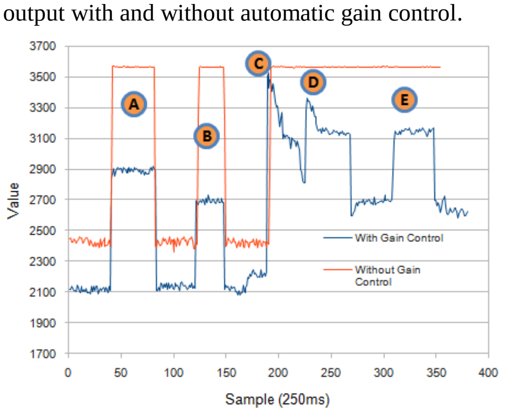


**Figure 7: Gain Control Reduces External Noise**

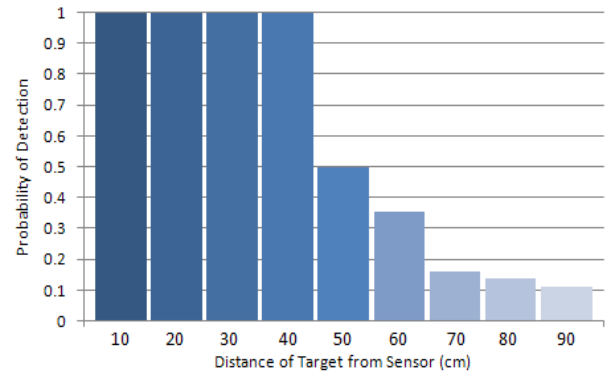


**Figure 8: Sensors are Accurate up to 40cm**

In our next test, seen in Figure 8, we measured QueueTrak's sensor accuracy at increasing distances by detecting an object at each distance from 10-90cm in 10cm increments. At distances less than or equal to 40cm, the sensors always reported the presence of a person. However, beyond this distance, accuracy quickly begins to fade. A sensing range of 40cm is sufficient for the target use case illustrated in Figure 1, where the area for the line is known to within about 1m, and a 40cm range is highly likely to detect a person within this area. Farther ranges could be achieved by transmitting at higher power, if necessary.

## 5. REFERENCES


[1] SenSource. (2009). People counting products. Retrieved April 14, 2009 from http://www.sensourceinc.com/peoplecounters.htm.

[2] St. Michael's Strategies. (2009). Small Chain Solution. Retrieved April 14, 2009 from http://www.storetraffic.com

[3] NICE Systems. (2009). NiceVision Analytics. Retrieved April 14, 2009 from http://www.nice.com/products/video/analytics.php.

[4] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. Irisnet: An architecture for a world-wide sensor web. IEEE Pervasive Computing, 2(4):22–33, 2003.

[5] J. Shneiderman, P. Pietzuch, J. Ledlie, M. Roussopolous, M. Seltzer, and M. Welsh. Hourglass: An infrastructure for connecting sensor networks and applications. Tech report, Harvard University, 2004.

[6] R. Dickerson, J. Lu, J. Lu, and K. Whitehouse. Streamfeeds: An abstraction for the world wide sensor web. In Internet of Things, 2008.

[7] S. N. J. Liu and F. Zhao. Challenges in building a portal for sensors world-wide. First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications (WSW'2006), 2006.

[8] Robert F. Dickerson Jiakang Lu Jingyuan Li Billy Chantree Jian Lu, John A. Stankovic, Kamin Whitehouse. MetroNet: Case Study for Collaborative Data Sharing on the WorldWide Web. IPSN '07.