

Caraoke: An E-Toll Transponder Network for Smart Cities

Omid Abari, Deepak Vasisht, Dina Katabi and Anantha Chandrakasan
Massachusetts Institute of Technology
Cambridge, MA, USA
{abari, deepakv, dk, anantha}@mit.edu

ABSTRACT

Electronic toll collection transponders, e.g., E-ZPass, are a widely-used wireless technology. About 70% to 89% of the cars in US have these devices, and some states plan to make them mandatory. As wireless devices however, they lack a basic function: a MAC protocol that prevents collisions. Hence, today, they can be queried only with directional antennas in isolated spots. However, if one could interact with e-toll transponders anywhere in the city despite collisions, it would enable many smart applications. For example, the city can query the transponders to estimate the vehicle flow at every intersection. It can also localize the cars using their wireless signals, and detect those that run a red-light. The same infrastructure can also deliver smart street-parking, where a user parks anywhere on the street, the city localizes his car, and automatically charges his account.

This paper presents Caraoke, a networked system for delivering smart services using e-toll transponders. Our design operates with existing unmodified transponders, allowing for applications that communicate with, localize, and count transponders, despite wireless collisions. To do so, Caraoke exploits the structure of the transponders' signal and its properties in the frequency domain. We built Caraoke reader into a small PCB that harvests solar energy and can be easily deployed on street lamps. We also evaluated Caraoke on four streets on our campus and demonstrated its capabilities.

CCS CONCEPTS

•Networks → Network protocol design;

KEYWORDS

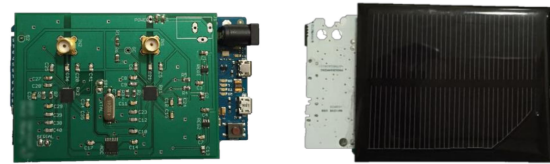
Wireless, Smart City, RF Localization, Active RFID, Electronic Toll Collection (ETC)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '15, August 17 - 21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-3542-3/15/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2785956.2787504>



(a) Bottom view

(b) Top view (solar panel)

Figure 1—Caraoke reader. The board is 3×4 in, and operates by harvesting solar power.

1. INTRODUCTION

Electronic toll collection transponders are simple devices consisting of a battery-powered RFID. They are perhaps among the most-widely used wireless communication technologies. In the US, depending on the state, 70% to 89% of the cars have such transponders [56, 2, 46, 9]. Further, the numbers are growing rapidly. The state of Pennsylvania has announced that E-ZPass will be mandatory on all highways in 2018. The state of California already requires drivers to have the transponder mounted on the windshield per state law in order to drive in the Express-Lanes [55, 9]. Other states are following suit motivated by Congress's decision to have a national electronic toll-collection system by 2016 [4, 9]. Because of this wide-deployment and anticipated growth, multiple businesses are looking into leveraging e-toll transponders to deliver new services. For example, e-toll transponders are currently used to pay for food at some drive-through restaurants [28], and to automate payment at parking garages [5].

More generally, there is a big opportunity for using e-toll transponders to enable smart cities. For example, the city could deploy readers on traffic lights to query the transponders and track the number of cars at every intersection. It can then use the information to adapt the timing of traffic lights to minimize the average wait time for the green light. It can also leverage RF-based localization to localize cars using their transponders' signals, detect cars that run a red-light, and automatically charge their accounts for a ticket. Readers deployed on street-lamps can detect speeding on every street in the city and ticket the offending car, without the need for car-mounted radars and hidden police officers. The same infrastructure can deliver smart street-parking systems, where a user parks anywhere on the street, the city localizes his car, and automatically charges his account.

Unfortunately, today there is a major challenge that hampers the use of e-toll transponders in smart city services, like the ones described above. Specifically, e-toll transponders are designed under the assumption that only one transponder transmits at any point in time, and hence have no MAC protocol to prevent collisions.¹ Collection systems use restricted deployments and highly directional antennas to ensure that only one car responds to the reader's query. Without this physical isolation, all transponders in range would transmit simultaneously, creating collisions. One could think of replacing the current transponders with new transponders that support a MAC protocol. Replacing the large infrastructure of deployed transponders however would take a long time and incur a major cost.² In contrast, developing for current transponders allows the cities to obtain immediate benefits even with a small installation on some of the busier streets and intersections. Yet to do that, the system has to deliver its smart services in the presence of wireless collisions.

This paper presents Caraoke, a networked system for delivering smart services using existing e-toll transponders. Caraoke also presents a new reader design that can count, localize, and estimate the speed of the cars on the road using collision signals from their e-toll transponders. The key feature that enables Caraoke to work in the presence of collision is its ability to exploit the carrier frequency offset (CFO) of the transponders. Specifically, since e-toll transponders are active RFIDs, each device has an independent oscillator, and hence it experiences a carrier frequency offset (CFO). Traditional wireless systems view the CFO as a harmful phenomenon that the receiver has to compensate for in order to correctly decode. In contrast, we show that we can leverage the CFO of the transponders to zoom in on individual transponders in the presence of collisions.

In particular, we consider the collision in the frequency domain as opposed to the time domain, and show that each collision exhibits spikes that correspond to the CFOs of the colliding transponders. Further, e-toll transponders have particularly large CFOs that span 1.2MHz [36], creating a significant separation between the spikes. Thus, we can estimate the number of transponders by counting these spikes.

We also show that we can use the differences in CFO to measure the wireless channels to the individual transponders, and hence apply RF-based localization to track cars and measure their speeds. Caraoke can also decode the IDs of the colliding transponders, say to charge a car for parking or speeding. To do this, Caraoke leverages the channels and CFO measurements to combine multiple collisions in a manner that the signals from the target transponder add up coherently, whereas the signals of other colliding transponders combine incoherently. This allows Caraoke to boost the SNR of the target transponder above the others, and enable it to decode the ID of the target transponder.

¹This is unlike traditional RFIDs used in access control or retail, which have a MAC protocol.

²There were more than 26 million transponders deployed just by E-ZPass as of 2013 [1]. In addition, there are a large number of additional transponders deployed by other agencies such as FasTrak, etc.

To demonstrate the practicality of Caraoke, we built Caraoke reader into a custom designed printed circuit board (PCB). Our prototype, shown in Fig. 1, is both small and low-cost, making it amenable to large-scale deployment.³ Further it is designed as a plug and play device; It connects to the Internet via an LTE modem and harvests its energy from solar power, making it easy to deploy on street-lamps.

We have evaluated Caraoke on four campus streets. We ran multiple experiments with cars that have standard E-ZPass transponders. Our results show the following:

- Caraoke can count transponders accurately despite collisions. The average error in the Caraoke estimator is 2%, and the 90th percentile is less than 5%, which is significantly more accurate than existing camera-based traffic tracking systems [43, 54].
- Caraoke can accurately localize cars into parking spots. Its average location accuracy is 4 degrees. This accuracy is sufficient for detecting occupied/available parking spots between two street lamps.
- Across experiments where we varied the car speed from 10 mile/hour to 40 mile/hour, Caraoke has detected the speed to within 8% (i.e., 1 to 3 mile/hour). The same accuracy was observed in a second set of experiments conducted in an empty lot with a car speed of 50 mile/hour.⁴
- Caraoke successfully decodes transponder ids in the presence of collisions, but the time required to decode increases with the number of colliding transponders. In particular, decoding the ids of a pair of colliding transponders takes 4.2 ms, whereas decoding five colliding ids takes 16.2 ms.
- Measurements of the Caraoke reader show that it consumes only 9mW in average (excluding modem), which is 56× lower than what it can harvest from its solar panel.

2. RELATED WORK

(a) Communication and Localization: Caraoke builds on a rich literature on RFIDs. Past research however has typically focused on EPC RFIDs, like those used in access control and inventory tracking [22, 30]. Such RFIDs *do* support a MAC protocol and hence can communicate without major collisions. In contrast, e-toll transponders use a different protocol that has no MAC support. We note however that past works [58, 34] proposed methods to decode concurrent transmissions from backscatter sensors in time domain. However, such designs are inapplicable to our scenario because they require hardware modification of the RFIDs and do not work with existing e-toll transponders.

Our work is also related to past work on RFID localization and RF-based positioning [60, 29, 59, 63, 26]. While we build on the general area of AoA localization, our approach

³ The board is about the size of a credit card and cost less than \$40 which can be dramatically reduced with mass production.

⁴Almost all states in the US have residential speed limits below 35 mile/hour, and the maximum residential speed limit in any state is 45 mile/hour [8].

differs in that it exploits CFO differences to localize the devices using colliding signals, without even decoding.

There are also commercial RFID readers which are solar powered [7]. However, unlike Caraoke, these readers can not localize, count and identify RFIDs in the presence of wireless collisions.

Finally, a vast majority of past research on issues related to CFO focuses on how to eliminate or estimate the CFO and compensate for it [12, 49, 48, 52]. The closest to our work is [18], which advocates using the CFO of a device as an id for security purposes. None of this work however deals with collisions or the use of CFO for localization or decoding.

(b) Smart Cities: Our work is motivated by the growing interest in smart cities, where urban services are automated to improve efficiency, and reduce waste and pollution [42, 35, 25]. Past work in this area focuses on transportation research [61], software applications [20], and social and economic issues [14]. In contrast, we focus on wireless networking issues such as communication, localization, and counting in the presence of wireless collisions.

There are also a few businesses that market solutions for one of Caraoke’s applications. In particular, some apps allow a user to pay for parking using her phone [19]. Those apps however do not address the cost and overhead incurred by the city in checking for parking violations and issuing tickets. Further, they do not automatically detect the occupancy of parking spots. Alternative solutions like Streetline install a sensor in the asphalt pavement of every parking spot [51]. They need to drill the street in every spot incurring a significant cost and causing traffic disturbances. There are also traffic cameras installed in some cities for counting the cars at the corresponding traffic light and providing traffic statistics [38]. These systems are highly sensitive to occlusions, illumination, shadowing, and wind [43]. Finally, traffic radars are typically used to measure car speeds. These devices however cannot tell which speed is associated with which car. A police officer has to be around to identify the speeding car based on the orientation of the device [24]. In contrast to all of the above, Caraoke is a single system that can support all of these applications, and address many of the drawbacks of existing solutions.

(c) VANET: There is a large literature on vehicle networks, or VANET. Research in that area addresses the impact of mobility on ad hoc networks [40]. It focuses on routing [39], quality of service [62], and reliability [64]. It runs on typical communication devices that support a MAC protocol, e.g., WiFi and WiMAX [21, 44]. Our work differs from this past work in objectives and techniques. Specifically, our goal is to enable smart cities to leverage the widely-deployed e-toll transponders to deliver new services such as smart-parking and real-time traffic monitoring. Our solutions target a different communication technology, namely e-toll transponders. Such transponders lack a MAC protocol, necessitating new designs that differ from those used in VANET.

3. BACKGROUND

An e-toll transponder is an active RFID, which responds to

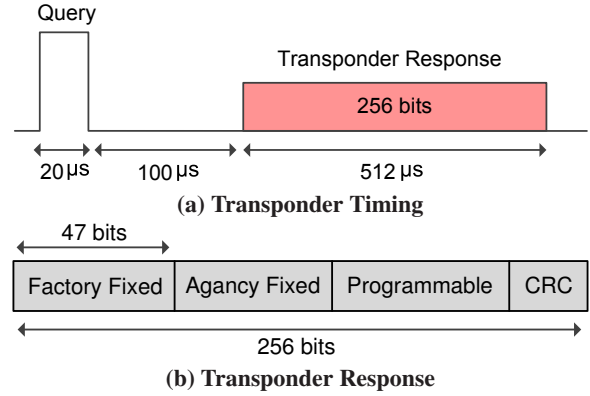


Figure 2—Transponder Transmission. The reader transmits a query signal which is a simple sine-wave at carrier frequency. The transponder responds by transmitting a 256-bit data using OOK Manchester modulation.

an inquiry transmitted by the reader. The reader is typically placed in the tollbooth whereas the transponder is attached to the car’s windshield. Both transponder and reader work at 915MHz. The query signal is simply a sinewave transmitted at the carrier frequency (i.e. 915MHz) for a short period of time. The transponder responds with its id, which identifies the driver’s account. Fig. 2(a) illustrates the timing of the query signal and the transponder’s response to it.

A few points are worth noting:

- In contrast to traditional RFIDs (e.g., those used in retail or access control),⁵ the active RFIDs used in e-toll transponders lack a medium access protocol (MAC). Thus, once a transponder detects the reader’s signal, it immediately transmits its response. Hence, if multiple transponders are in the reader’s range, they all respond leading to a collision. Toll systems avoid the need for a MAC by using highly directional antennas, and the fact that cars are separated by a minimum distance.
- E-toll transponders also have a relatively large CFO. Their carrier frequencies vary between 914.3MHz and 915.5 MHz, and hence their CFO can be as high as 1.2MHz [36].
- The simplicity of the transponders results in a cheap and low power device. A transponder can work for 10 years before it runs out of battery, and it operates whether the car is on or off.
- Finally, for the purpose of this paper, it is important to understand the properties of the transponder signal. The transponder transmits its data using on-off keying (OOK) modulation. OOK is a simple modulation, where the radio transmits a “1 bit” by transmitting the carrier frequency, and transmits a “0 bit” by staying silent. This means that the transponder’s signal corresponds to the presence and absence of the carrier sinewave. Thus, the transmitted signal can be written as:

$$x(t) = s(t) \cdot e^{j2\pi \cdot f_c \cdot t}, \quad (1)$$

where $s(t)$ is a binary square-wave baseband signal toggling between 0 and 1, and f_c is the carrier frequency. The

⁵Most research targets Electronic Product Code (EPC) RFIDs, which have an Aloha-style MAC protocol [23].

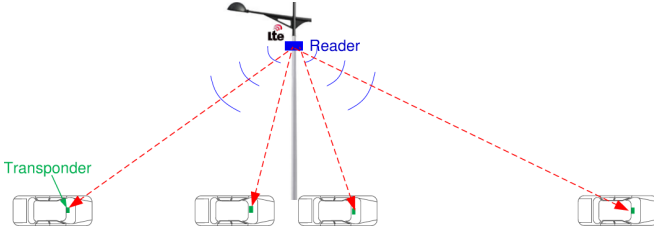


Figure 3—Illustration of Caraoke The Caraoke reader is mounted on a street-lamp. It queries nearby transponders and uses their signals to localize them and estimate their speeds.

received wireless signal can be written as:

$$y(t) = h \cdot s(t) \cdot e^{j2\pi \cdot f_c \cdot t}, \quad (2)$$

where h is the complex channel coefficient. The receiver down-converts the signal to baseband by multiplying it with its own carrier frequency. The received baseband signal $r(t)$ then becomes:

$$r(t) = h \cdot s(t) \cdot e^{j2\pi \cdot \Delta f \cdot t} \quad (3)$$

$$= h \cdot (0.5 + s'(t)) \cdot e^{j2\pi \cdot \Delta f \cdot t}, \quad (4)$$

where Δf is the carrier frequency offset between the transmitter and the receiver, and $s'(t)$ is the same square-wave as $s(t)$ except that it toggles between -0.5 and 0.5 and has zero mean. The frequency representation of the received signal $r(t)$ can be written as:

$$R(f) = \frac{h}{2} \cdot \delta(f - \Delta f) + h \cdot S'(f - \Delta f) \quad (5)$$

where $S'(f)$ is the frequency representation of $s'(t)$ and $\delta(f)$ is the unit impulse function. As it can be seen from the equation, this signal has a peak at the carrier frequency offset, Δf . Further, since $s'(t)$ has a zero mean, $S'(0) = 0$.⁶ Thus, the complex value of the peak represents the channel from transmitter to receiver *i.e.*, $R(\Delta f) = \frac{h}{2}$.

4. CARAOKE OVERVIEW

Caraoke is a networked system that enables query-response communication between a Caraoke reader and the e-toll transponders in its range. At the heart of Caraoke is a new device that we call the Caraoke reader; it counts, localizes, and decodes transponders' ids from their signal collisions. It also estimates the speeds of the cars carrying the transponders. The Caraoke reader harnesses its power from solar energy and has an LTE modem to connect to the Internet. Hence, it can be easily deployed without the need for additional infrastructure. As shown in Fig. 3, smart cities can deploy Caraoke readers on street-lamps to support a variety of smart services including: 1) traffic monitoring, 2) speed enforcement, 3) red-light running, 4) smart street-parking, and even 4) allowing a user who forgets where he parked to query the system to locate his parked car.

⁶ $s'(t)$ has a zero mean because $s(t)$ is an on-off keying signal with Manchester encoding, and $s'(t)$ is the same as $s(t)$ but shifted by -0.5.

This paper is focused on the design and implementation of the Caraoke and a small-scale evaluation of the deployment of multiple Caraoke readers on a campus street. Before delving into the details of our design, we note the following two points regarding scope:

- Our objective is to automate smart services, eliminating the personnel cost, and improving the overall accuracy in comparison to the status quo. Note that the current alternatives suffer from significant errors. For example, about 10% to 30% of the speeding tickets based on traffic radars are estimated to be incorrect [6]. The errors are mostly due to the fact that radars cannot associate a speed with a particular car. This task is left to the police officer and hence is prone to human mistakes [6]. Similarly, errors in estimating the number of cars using traffic cameras vary between a few percent to 26%, depending on illumination, wind, occlusions, etc. [43]. Furthermore, the camera lenses have to be manually cleaned every 6 weeks to 6 months [16].
- For a city to use Caraoke to deliver the above services, it needs to connect the system with its own transportation and traffic databases. For example, in order to detect a car that runs a red light, the city needs to combine the output of Caraoke with the timing of the red-light at the corresponding intersection. The process for combining Caraoke's output with the city's transportation and traffic databases is beyond the scope of this paper.

5. COUNTING DESPITE COLLISIONS

Estimating the number of vehicles at major intersections is critical for traffic management and city planning. In this section, we describe how a Caraoke reader counts the transponders in its radio range. Specifically, when a Caraoke reader transmits a query message, transponders in its radio range respond simultaneously with their information. We would like to use the resulting collisions to count the number of transponders.

At a high level, our approach is simple. We exploit the fact that two transponders, typically, do not have the same carrier frequency, and that their carrier frequency offset (CFO) is relatively large. In particular, the specifications of the E-ZPass transponder show that the device's CFO can exceed one MHz [36]. CFO is typically a nuisance for wireless communication systems which have to compensate for CFO before decoding. In Caraoke however, we leverage CFO for our advantage to count the number of colliding transponders. Specifically, we take the FFT of the collision signal. Since different transponders have different carrier frequencies, the Fourier transform shows multiple peaks at different frequencies that corresponds to the various transponders' CFOs. Fig. 4 shows the Fourier transform of a collision signal where five e-toll transponders transmitted at the same time. As can be seen in the figure, there are five peaks, each corresponds to one of five colliding transponders.

This shows that one way for counting the transponders would be to take an FFT of the collision signal and count the peaks in the Fourier domain. To understand the performance of this estimator, we need to tie it to the resolution of the

FFT and whether it can distinguish the differences between the CFOs of the transponders. The resolution of the FFT, δf refers to the width of each FFT bin and can be written as:

$$\delta f = \frac{1}{T} \quad (6)$$

where T is the FFT time window. Since the length of the transponder's response is $512\mu s$, the maximum FFT window is $T = 512\mu s$ and hence the resolution of the FFT is $\delta f = 1.95$ kHz. Thus, if two transponders have carrier frequencies that differ by less than 1.95 kHz, their peaks will fall into the same FFT bin and will be counted as one. Given that the CFO range is 1.2 MHz, the peak of a transponder can fall in any of $N = 1.2 \text{ MHz}/1.95 \text{ kHz} = 615$ FFT bins. If m transponders collide, then the probability of not missing any transponder by counting FFT peaks is:

$$P(\text{not missing any transponder}) = \frac{\binom{N}{m} \cdot m!}{N^m} \quad (7)$$

Unfortunately this probability decreases quickly as more transponders are in range. The probability of not missing any transponder is 98%, 93% and 73% for $m = 5, 10$ and 20 cars, respectively. The above derivations shows that an estimator that simply counts the number of peaks in the FFT is acceptable at low car density but can easily miss some cars when the number of cars in range is large.

So, how can we improve the quality of our estimate in scenarios of high densities? To overcome this problem, Caraoke distinguishes whether one or more transponders have fallen into the same FFT bin while counting the number of peaks. It does this by leveraging the *phase rotation* property of the Fourier transform, which says that a shift in the time domain translates into phase rotation in the frequency domain:

$$\begin{aligned} \mathcal{F}\{r(t)\} &= R(f) \\ \mathcal{F}\{r(t + \tau)\} &= R(f) \cdot e^{j2\pi \cdot f \cdot \tau} \end{aligned} \quad (8)$$

where $r(t)$ is the signal in time domain and $R(f)$ is its frequency representation. Specifically, if the FFT peak contains a single transponder's response, then performing the FFT on the same signal with a time shift τ causes only a phase rotation of the peak value but the magnitude of the peak does not change. In other words, $\|R(f)\| = \|R(f) \cdot e^{j2\pi \cdot f \cdot \tau}\|$, where $R(f)$ is the frequency representation of the received signal. In contrast, say the CFOs of two transponders, f and f' , fall into the same FFT bin, then the value of the peak in that bin without a time-shift is $R(f) + R(f')$ while its value with a time-shift of τ is $R(f) \cdot e^{j2\pi \cdot f \cdot \tau} + R(f') \cdot e^{j2\pi \cdot f' \cdot \tau}$. Since the frequencies are slightly different, they rotate by different phases and results in a change in the magnitude of the peak.

The above provides us with a mechanism to determine whether an FFT bin has one or more transponders. To do so, we compare the magnitude of the FFT bin with and without a time-shift. If the two magnitudes are different by more than a noise threshold, then multiple transponders have fallen into that bin. In the following, we explain how this detection significantly improves the probability of getting a correct count.

Probability of getting the correct count: As explained before, Caraoke counts the number of the peaks in the FFT to

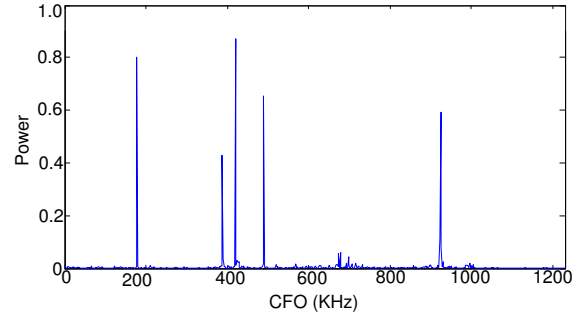


Figure 4—Transponders Response in Frequency Domain. Fourier transform of a collision signal where five e-toll transponders transmitted at the same time. The number of peaks indicates the number of transponders in the area.

determine the number of cars in range. However, there is a possibility that two cars have fallen into the same bin. Hence, Caraoke considers the peaks with two or more transponders' signals as two cars when it is counting the peak. Specifically, if an FFT peak includes a single frequency, Caraoke counts it as one car and if it has two or more frequencies, it counts it as two cars. Hence, the result of counting will be incorrect only when there is at least a bin which includes three or more cars. In another word, the probability of not missing any transponder is equal to one minus the probability of having at least one FFT bin which includes three or more transponders' signal. For m colliding transponders and N FFT bins in the 1.2 MHz range, this probability becomes:

$$\begin{aligned} P(\text{not missing any transponder}) &= 1 - P(\exists \text{ bin with } \geq 3 \text{ transponders}) \\ &\geq 1 - \sum_{i \in \{1, \dots, N\}} P(\text{bin } i \text{ with } \geq 3 \text{ transponders}) \quad (9) \\ &\geq 1 - \binom{N}{1} \binom{m}{3} \frac{N^{(m-3)}}{N^m} \end{aligned}$$

Substituting $N = 615$, the probability of not missing any transponder is at least 99.9%, 99.9% and 99.7% for $m = 5, 10$ and 20. Thus, having the capability to detect the peaks that have two or more transponder's signals significantly improves the probability of correctly counting the cars.

Finally, note that for simplicity, our analysis has assumed a uniform distribution for CFO. However, we have also experimentally validated our solution for empirical CFO measurements collected from 155 different transponders.⁷ Our empirical results show that the probability of not missing any transponder is 99.9%, 99.5% and 95.3% for $m = 5, 10$ and 20 which are slightly worse than analytical-results.

6. LOCALIZING E-TOLL TRANSPONDERS

Car localization is an essential function for multiple smart services such as smart parking and detecting red-light runners. The first step in localizing cars is to localize the transponder located on cars' windshield. To do so, Caraoke first calculates the angle at which the signal from

⁷The mean and standard deviation of their carrier frequencies are 914.84 MHz and 0.21 MHz, respectively.

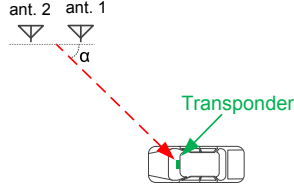


Figure 5—Transponder Localization. Caraoke calculates the spatial angle (α) of transponder by measuring the phase difference in signal received at two antennas.

the transponder arrives at the Caraoke reader. This angle indicates the spatial angle between the transponder and reader, and typically referred to as the angle-of-arrival (AoA).

In order to measure the AoA, a Caraoke reader uses two antennas separated by a distance d . It is widely-known that the angle of arrival can be computed as [60]:

$$\cos \alpha = \frac{\Delta \phi \lambda}{2\pi d}, \quad (10)$$

where α is the spatial angle between the transponder and the reader, shown in Fig. 5, $\Delta \phi$ is the phase difference between the two antennas (*i.e.* $\Delta \phi = \phi_2 - \phi_1$), and λ is the carrier wavelength⁸.

To compute α , we need to substitute the value of the other parameters in Eq. 10. While d and λ are known, $\Delta \phi$ should be measured. In the absence of collisions, $\Delta \phi$ can be measured directly between the signals received by the two antennas on the reader:

$$\Delta \phi = \angle \frac{r_2(t)}{r_1(t)} = \angle \frac{h_2 \cdot s(t) \cdot e^{j2\pi \Delta f \cdot t}}{h_1 \cdot s(t) \cdot e^{j2\pi \Delta f \cdot t}} = \angle \frac{h_2}{h_1},$$

where $r_1(t)$ and $r_2(t)$ are baseband signals received by the first and second antenna, respectively, h_1 and h_2 are the channels to the two antennas, $s(t)$ is the transmitted baseband signal and Δf is the CFO between the transmitter and receiver⁹. However, since in Caraoke reader each antenna receives a collision from multiple transponders, such a direct measurement does not work. This is due to the fact that the received signal is the summation of responses from multiple transponders while each has its own channel. Thus, the received signal at each antenna when m transponders respond can be written as:

$$\begin{aligned} r_1(t) &= r_{11}(t) + r_{12}(t) + \dots + r_{1m}(t) \\ r_2(t) &= r_{21}(t) + r_{22}(t) + \dots + r_{2m}(t), \end{aligned} \quad (11)$$

where $r_{1i}(t)$ and $r_{2i}(t)$ are the received signal from the i_{th} transponder to the first and second antenna, respectively. As

⁸Note that since Caraoke reader is placed outdoor on a high pole like a street lamp, it has a strong line-of-sight path to the transponder and the multipath effects which occur in standard indoor environments are significantly weaker. Hence, Caraoke does not require a large antenna array to estimate the angle of arrival. In §12, we empirically show that Caraoke does not suffer from multipath effects.

⁹Note that the received signals on the two antennas experience the same CFO since the antennas are connected to the same oscillator on the Caraoke reader.

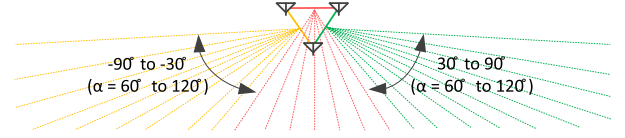


Figure 6—Antenna Re-configuration. Caraoke reader has three antennas arranged in an equilateral triangle to achieve higher accuracy in calculating spatial angle of transponder. In this case the spatial angle, α , is always between 60 to 120 degrees regardless of transponder location

can be seen from the equations, one can not directly compute $\Delta \phi$ for the i_{th} transponder using $\angle \frac{r_2(t)}{r_1(t)}$ since $\angle \frac{r_2(t)}{r_1(t)} \neq \angle \frac{h_2}{h_1}$ for a specific transponder.

Fortunately, however, we can use the same trick we used for counting the transponders. Specifically, we first take the FFT of the collision at each antenna and identify the peaks, where each peak corresponds to the response from one transponder. For each peak in the first antenna's signal, the phase value is compared to the phase value of the same peak in the other antenna's signal. These phase differences are used to calculate the spatial angle α for each transponder. Mathematically, the above approach works because the Fourier transform is linear, *i.e.*:

$$\mathcal{F}\{ax(t) + by(t)\} = aX(f) + bY(f) \quad (12)$$

Using the above property, the frequency representation of the received signal at the antennas when m transponders respond can be written as:

$$\begin{aligned} R_1(f) &= R_{11}(f) + R_{12}(f) + \dots + R_{1m}(f) \\ R_2(f) &= R_{21}(f) + R_{22}(f) + \dots + R_{2m}(f) \end{aligned} \quad (13)$$

where $R_{1i}(f)$ and $R_{2i}(f)$ are frequency representation of the received signals from the i_{th} transponder to the first and second antenna, respectively. As it was explained in §3, the received signal from each transponder has a peak at its CFO where the value of the peak represents the channel coefficient (*i.e.* $R(\Delta f_i) = \frac{h_i}{2}$). Hence, $R_1(f)$ and $R_2(f)$ signals have multiple peaks where each peak corresponds to the response from only one transponder. Therefore, the $\Delta \phi$ for the transponder i can be calculated as follow:

$$\Delta \phi = \angle \frac{R_2(\Delta f_i)}{R_1(\Delta f_i)} = \angle \frac{h_{2i}}{h_{1i}}$$

where Δf_i is the CFO of transponder i , and $R_2(f)$ and $R_1(f)$ are frequency representation of the received signal at the first and second antennas. Substituting the measured $\Delta \phi$ in Eq. 10, we can compute the spatial angle between the transponder and reader (*i.e.* AoA).

The above equation allows us to compute the spatial angle from the reader to the transponder. We can however improve the accuracy of our angle estimate with a smart choice of antenna position. Specifically, the accuracy in calculating α is best for angles around 90° and degrades for angles around 0° or 180° . This is due to the fact that $\Delta \phi$ is proportional to $\cos \alpha$, as shown in Eq. 10. Hence, for values close to 0 or 180, α is very sensitive to change in $\Delta \phi$. To reduce this sensitivity, we use three antennas arranged in an equilateral triangle as shown in Fig. 6. At any time, we use a pair of

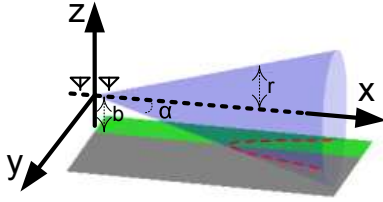


Figure 7—Localizing the Car from AoA: In this figure α is angle of arrival and b is the height of the pole. The car is located on a point on the intersection of the cone and the road plane which is a hyperbola. Hence, By using the information from two poles, one can localize the exact location of the car.

antennas out of the three antennas. We pick the pair using a programmable switch. In this setup, for any transponder position, there exists exactly one pair of antennas for which the spatial angle is always close to 90° (i.e., between 60° and 120°). We compute the angle for all pairs and use the pair whose angle is close to 90° degree to localize the car.

Next we use the spatial angle to locate the transponder. The spatial angle does not correspond to a single point in the space. In fact, as shown in Fig. 7, this angle corresponds to all points on the surface of a cone where its altitude axis is parallel to the road. The cone equation can be written as follow:

$$y^2 + z^2 = r^2 = (\tan(\alpha) \cdot x)^2 \quad (14)$$

where x, y and z are coordinate of the car with respect to the center of measuring antennas, and α is the spatial angle between the car and reader. Cars, however, are always on the road and we can use this as another constraint. The intersection of the cone and the road plane is a hyperbola, as shown in Fig. 7. The equation for this hyperbola is as follow:

$$(\tan(\alpha) \cdot x)^2 - y^2 = b^2 \quad (15)$$

where b is a constant that corresponds to the height of the pole. While a single hyperbola is not enough to localize, we can combine information across two readers to locate the car. Specifically, using a second reader located on the other side of the road provides us another hyperbola equation. Then, by solving these two equations, one can find x and y , and localize the car.¹⁰

Note, in the case where antennas are tilted by 60° , the process of localizing is the same, except that the cone is tilted by 60° . Hence, the intersection of the cone and road plane is an ellipse instead of a hyperbola. One can simply replace the hyperbola equation with that of an ellipse to find the intersection point.

Finally, in order to intersect location information across two readers, we need this information to be synchronized. We can leverage the readers' connection to the Internet to synchronize them to within tens of ms using the network timing protocol (NTP) [3, 45]. This synchronization level is more than sufficient for localizing parked cars. For moving

¹⁰ The intersection of two hyperbolas may results in more than one point, however, only one of these points is located on the road and the rests are on the sidewalk.

cars, this introduce some error, which we will discuss in the following section.

7. DETECTING A CAR'S SPEED

As described earlier, Caraoke can also detect the speed of the car. Specifically, Caraoke readers can be deployed on street-lamps and detect speeding on streets in the city. The car speed can be estimated by localizing the car at two different locations and computing the total time the car took to travel between these two locations. Hence, the speed of the car can be written as:

$$v = \frac{x_2 - x_1}{delay}$$

where x_1 and x_2 are first and second locations of the car and $delay$ is the amount of time it took to travel from location x_1 to location x_2 which are computed as described §6.

The accuracy of localizing the x_1 and x_2 and estimating the $delay$ depends on the time synchronization between the readers. The error in x_1 and x_2 can be upper bounded using the hyperbola equation in the previous section independent of time synchronization. This error depends on reader's height and the number of lanes in the same direction on the street. For example, for a four lane street *i.e.* two lanes in each direction, where the antennas are attached to a street light pole whose height is 13 feet, the maximum error is 8.5 feet.¹¹ The error in $delay$ is the same as the error in timing synchronization. Since the readers are connected to the Internet via LTE modems, they can be synchronized up to tens of ms network timing protocol (NTP) [3, 45].

The accuracy of estimating the speed depends on the accuracy of the above parameters as well as how far x_1 and x_2 are from each other. The farther they are, the more accurate the estimate is. In particular, if x_1 and x_2 are measured at readers that are separated by 4 light poles (*i.e.* a separation of about 360 feet (≈ 110 m) [10]), for car speeds of 20 mile/hour and 50 mile/hour, the maximum error is 5.5% and 6.8% respectively. This accuracy can further be improved by taking more measurements along the street from more light poles.

8. DECODING TRANSPONDERS' IDS

In this section, we explain how Caraoke decodes an individual transponder in the presence of collisions of multiple transponders.

At first glance, it might seem that one can decode a transponder's signal by using a band-pass filter centered around the transponder's CFO peak. This solution however does not work because OOK has a relatively wide spectrum *-i.e.*, the data is spread as opposed to being concentrated around the peak¹².

¹¹The exact equation of error is $\frac{\sqrt{b^2 - \sqrt{b^2 + (l \cdot w)^2}}}{\tan(\alpha)}$ where b is antenna's height, l is the number of lanes in the same direction on the street and w is the width of the lane (typically 12 feet).

¹²Intuitively this can be seen by recalling that OOK randomly toggles between 0 and 1 and hence it's spectrum resembles white noise.

In contrast, our decoding algorithm is based on combining multiple collisions in a manner that ensures that the signal from the target transponder combines coherently, whereas the signals from other transponders combine incoherently. This allows Caraoke to boost the SNR of the target transponder above others, and hence decode the target transponder.

Specifically, when a Caraoke reader transmits the query signal, multiple transponders respond simultaneously. Without loss of generality, let us assume that we are interested in decoding transponder 1.

$$r(t) = h_1 s_1(t) \cdot e^{j2\pi \cdot \Delta f_1 \cdot t} + \sum_i h_i s_i(t) \cdot e^{j2\pi \cdot \Delta f_i \cdot t}$$

where $s_i(t)$ is the signal transmitted by the i_{th} transponder, Δf_i is its CFO, and the h_i is its channel to the reader. If the reader transmits another query, the received signal will be:

$$r'(t) = h'_1 s_1(t) \cdot e^{j2\pi \cdot \Delta f_1 \cdot t} + \sum_i h'_i s_i(t) \cdot e^{j2\pi \cdot \Delta f_i \cdot t}$$

Note that the channel coefficients have changed from the first received signal to the second one. This is due to the fact that the transponders start with a random initial phase. The channels h_1 and h'_1 as well as the CFO Δf_1 can be estimated from the peak in the frequency domain as described in §3. We can then compensate for the CFO and the channels of transponder 1 and sum up the received signals to obtain the averaged signal $\tilde{s}_1(t)$:

$$\begin{aligned} \tilde{s}_1(t) &= \frac{r(t)}{h_1} \cdot e^{-j2\pi \Delta f_1 t} + \frac{r'(t)}{h'_1} \cdot e^{-j2\pi \Delta f_1 t} \\ &= 2 \cdot s_1(t) + \sum_i \left(\frac{h_i}{h_1} + \frac{h'_i}{h'_1} \right) s_i(t) \cdot e^{j2\pi (\Delta f_i - \Delta f_1) t} \end{aligned}$$

By repeating this process N times we get:

$$\tilde{s}_1(t) = N \cdot s_1(t) + \sum_i \left(\sum_j \frac{h_{ij}}{h_{1j}} \right) s_i(t) \cdot e^{j2\pi (\Delta f_i - \Delta f_1) t}$$

where h_{ij} is the channel from the i -th transponder in the j -th received signal. As can be seen from the above equation, the signals from transponder 1 add coherently, while the other signals add incoherently with random phases and average out. For sufficiently large N , the signal power for transponder 1 will be much more than that for other transponders. In this case, the SNR is enough to be decoded.

Fig. 8 shows an example of this decoding algorithm in which the reader receives a collision of the signals from five transponders. Fig. 8(a) shows the time signal $r(t)$ of the received collisions before any averaging. As can be seen, the signal looks random and undecodable. Fig. 8(b) and (c) show the time signal $\tilde{s}_1(t)$ after averaging 8 and 16 replies respectively. The figures show that after averaging 16 times, the bits of the desired transponder become decodable and the more we average, the better our ability to decode becomes.

9. CARAOKE MULTIPLE READER PROTOCOL

So far, we have assumed that transponders respond to a single reader at any time. However, a transponder on the

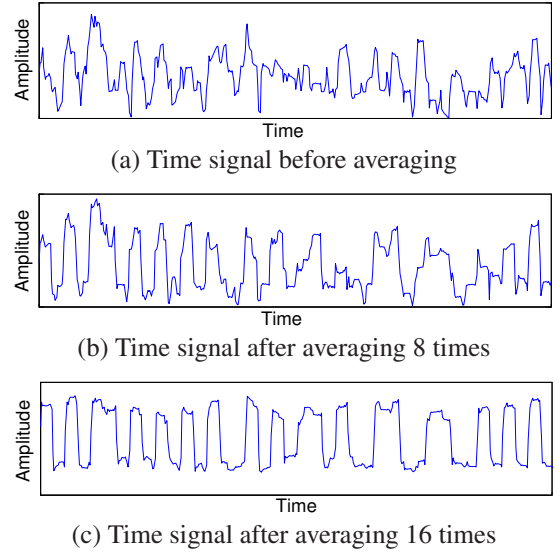


Figure 8—Decoding a transponder’s response: (a) Received signal before averaging, (b) after averaging 8 replies, (c) after averaging 16 replies. Caraoke is able to decode a transponder’s response by correcting its channel and then averaging.

road might be in the range of two or more Caraoke readers.¹³ Therefore, there is a need for a MAC protocol on the readers side to avoid interference from readers. We will start by distinguishing between two interference scenarios:

1. Collision of Reader Queries: In this case, a query signal from a reader collides with a query signal from another reader. As explained in §3, the query signal transmitted by a reader is simply a sinewave transmitted at carrier frequency. Even if two readers interfere, the combined signal is still a sinewave at the carrier frequency, and hence a valid trigger. Thus, a collision of two queries is not harmful. Our empirical experiments confirm that transponders are still triggered to respond even when queries from different readers collide.

2. Collision of Reader Query with a Transponder Response: In this case, a query signal from a reader collides with the response of a transponder queried by another reader. This collision is harmful and needs to be avoided. To do so, Caraoke uses carrier sense. Specifically, each reader listens to the medium before transmitting a query. If the medium is available, it then transmits its query. *But how long should the reader listen to avoid a collision with a transponder response.* Recall from Fig §2 in §3, that the query signal is only $20\mu s$ and the delay between the query and the transponder response is $100\mu s$. Thus, by listening for more than $120\mu s$, if the reader does not hear any signal it can guarantee that no transponder response will be transmitted after the $120\mu s$ and it can transmit its query.

To summarize, Caraoke uses a MAC protocol for the readers based on CSMA, where each reader listens for an idle medium for $120\mu s$ before it can transmit. The main difference, however, is that there is no need for contention window since collisions between queries are acceptable.

¹³The range of a Caraoke reader is 100 feet.

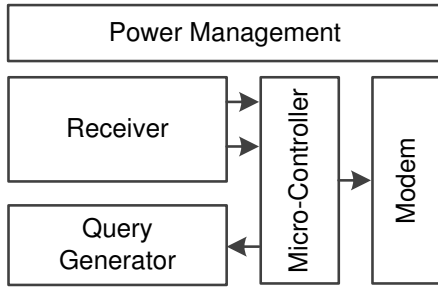


Figure 9—Block diagram of the Caraoke reader

10. CARAOKE READER’S HARDWARE DESIGN

Caraoke is a software-hardware solution. In contrast to the previous sections, which focus on the algorithmic techniques underlying Caraoke, here we describe the hardware design and the optimizations we performed in order to support a low-power low-cost device.

We have developed a custom-design PCB for the Caraoke reader. The device harvests its power from solar energy and connects to the Internet via a wireless modem. Hence, it can be attached to a light pole (or other structures) without the need for external power or wired Internet connectivity.

Fig. 9 illustrates the block diagram of the Caraoke reader. The device has five main block: a query generator, a receiver, a micro-controller, a modem, and a power management unit. The *query-generator* transmits a query signal that invokes a response from nearby transponders, the *receiver* receives the response signal, digitizes it, and provides it to the micro-controller. The *micro-controller* implements the standard receiver processing, i.e., packet detection, phase estimation, etc. It also implements our algorithmic solutions for counting, localization, speed measurement, etc. The processed data is then uploaded to the Internet via an LTE modem. (An alternative approach could use a WiFi modem, and have the readers forming a mesh network to connect to the Internet. However, the mesh network formation is beyond the scope of this paper.) Finally, the *power management* unit includes a small solar panel that attaches on top of the PCB, and a rechargeable battery that stores the harvested energy for operation during night time and on cloudy days. Solar panels can harvest $\sim 10mW/cm^2$ on sunny days [57, 53]. We use a 6cmx7.5cm solar panel, which can provides 500mW [47]. In §12, we show that this number is several folds larger than the power consumption of a Caraoke reader.

Finally, the device is equipped with a USB port which is used to program the micro-controller. The same USB port can also be used to power the device and extract the data, during the development phase.

In order to minimize the cost and power consumption of Caraoke reader, we perform multiple optimizations as follows:

- **Eliminating the TX chain:** A Caraoke reader has to transmit a query signal to request a response from the transponders. One approach to generate this query is to use a standard transmitter (TX) chain, which generates the signal in the digital domain. This approach however is in-

efficient in terms of both power and cost. Specifically, a full TX chain includes a mixer, a frequency synthesizer (PLL), a digital-to-analog-converter (DAC) and some digital logic. These components are necessary to transmit data bits. Yet, the e-toll query signal is just a simple sine-wave and does not carry any data. Such a sine-wave can be generated using a simple frequency synthesizer (i.e., a PLL), and does not need the rest of the components in the standard TX chain. The frequency synthesizer is turned on/off by the micro-controller so that the resulting sine-wave has the proper duration of the e-toll query.

- **Replacing FFT with the Sparse FFT:** As mentioned in §6, Caraoke operates on the response signal in the frequency domain after taking an FFT. At any time, the number of transponders that respond to the query is typically much smaller than the FFT size. In fact in most cases only a handful of transponders respond. Remember that each transponder creates a peak at the output of the FFT. Thus, the output of Fourier transform is sparse. For such sparse signals, one can compute the Fourier transform in sub-linear time [32, 31]. To do so, we use the sFFT algorithm in [33, 11], which is both simple and has low computation complexity. This directly translates into reduction in power consumption and speedup in the computation.
- **Duty Cycling:** Caraoke reader has two operating modes: active mode and sleep mode. In the active mode, the reader transmits queries, receives responses, and processes the signals. In the sleep mode, the query generator and receiver are turned off by the micro-controller by shutting off their power supply. All peripheral blocks are also deactivated. However, the master clock continues to run. A timer (typically called sleep timer) uses this clock to wake the micro-controller up at a specific time. The micro-controller controls the scheduling of the active and sleep modes to satisfy a desired power budget. Since each query takes about 1ms, we expect the average duration of the active mode to last for 10ms, allowing for a maximum of 10 queries. By controlling the duration of the sleep mode, the total power consumption can be dramatically reduced as we show in §12.5.

11. EVALUATION SETUP

Implementation: We implemented Caraoke reader on a printed circuit board (PCB) using off-the-shelf components. For the RF front-ends we use Maxim Integrated MAX2117 [41] and Analog Device AD7356 [15], and the Calliope LTE modem [50]. For the micro-controller, we use an Arduino Due board. The power management circuit includes regulator to regulate the voltage from the solar panel. For the solar panel, we use OSEPP SC10050 [47]. The RF chains share the same clock for accurate synchronization. The antennas are omni-directional and separated by $\frac{\lambda}{2} = 6.5 inches$. The ADC resolution is 12 bits. Also, the ADCs have differential inputs, and hence higher robustness to noise and interference. The total cost of the components is less than \$40, making the device amenable to large scale deployment. Further, this cost can be significantly reduced in mass production.

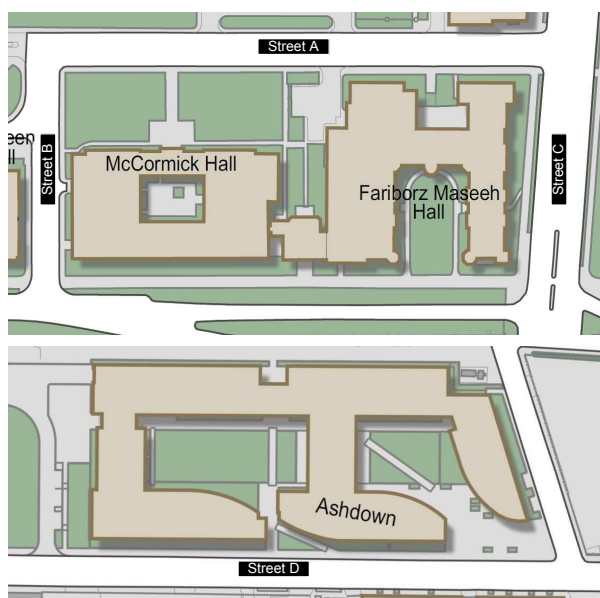


Figure 10—Experimental Setup. We ran our experiments on four streets in our campus, marked as streets A, B and C in the top figure and street D in the bottom figure. All streets have two-way traffic and most have parking spots on one or both sides.

We note that for our setup, we extract the data using the USB port as this does not require an LTE subscription and has no impact on counting, localization, or speed estimation.

Experimental Setup: We conducted our experiments on four campus streets, A, B, C, and D, which are shown in Figs. 10(a) and (b). All streets have 2-way traffic. Streets A, B, and D have street parking one or both sides of the road. Street C is the busiest street on campus, and is a major street in our city. In all of our experiments, the Caraoke reader was placed on a 12.5-foot pole. We used a total of 4 such poles. The poles are portable and hence allow us to experiment with various configurations, as detailed in §12.2.

All experiments were conducted with standard E-ZPass transponders attached to the cars’ windshields. We tried to limit our experiments to transponders and cars owned by the authors. However, certain experiments require investigating the distribution of transponder CFOs and its impact on our ability to count vehicles. For that experiment, we collected transponder responses from random cars. We measured only the CFO of the transponders and did not decode the bits. After processing the signals to extract and count the CFOs, we stored only the CFO values with no reference to the ids of the car. We do not believe that the values of the CFOs can be mapped to the owners or used to infer any private information about them. Finally, we note that our transmissions of a sine-wave in the band used by E-ZPass is in accordance with the FCC rules Part 15 [27].

12. EMPIRICAL EVALUATION OF CARAOKE

We evaluate the various functions of Caraoke using outdoor experiments performed with E-ZPass transponders.

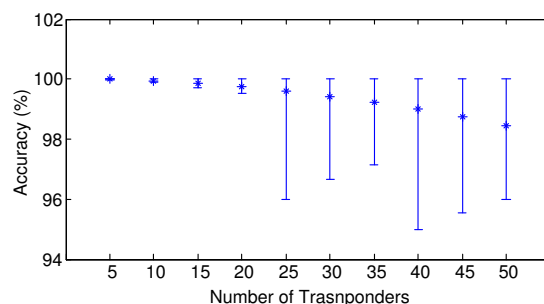


Figure 11—Counting Accuracy: Plot of Caraoke’s counting accuracy as the number of colliding transponders increases.

12.1 Counting Accuracy

In the first experiment, we aim to evaluate Caraoke’s ability to count transponders based on their CFOs. If we ran this experiment directly on collision signals, we would not know the ground truth. Hence, we needed first to estimate the CFO of each transponder in the absence of collisions. We collected signals from 155 different transponders in one of our campus parking lots. We used a directional antenna to obtain the response of each transponder without collisions. For each transponder, we took the FFT of its signal and noted the FFT bin of its CFO. We then create collisions in post-processing by summing up the time signals from a subset of the transponders. We change the number of transponders in the subset to obtain collisions with different numbers of colliding transponders. Finally, we take an FFT of each collision signal and estimate the number of colliding transponders using the approach described in §5. We have considered collisions of 5, 10, . . . , 50 transponders, and for each case performed 1000 runs.

Fig. 11 plots the average accuracy in counting colliding transponders. The figure shows that when the number of transponders in a collision is relatively small, Caraoke accuracy is very close to optimal. In particular, given the empirical CFO values of e-toll transponders, Caraoke can maintain an accuracy higher than 99% when the colliding transponders are fewer than 40. Note that the overall counting accuracy depends on how often the reader would experience 40-transponder collisions vs. 5-transponder collisions, or other numbers. This depends on the intersection and the amount of traffic. Overall, the results show that Caraoke is effective at counting vehicles using e-toll transponder collisions.

In our second experiment, we deployed Caraoke at the intersection of Street A and Street C in Fig. 10(a), and used it to track the flow of traffic on the two streets. In this experiment, we do not know the ground truth since we do not know which cars have transponders; despite this limitation the results can indicate the ratio of traffic between the two streets and how it relates to the timing of their green-light and red-light.

Fig. 12 plots the number of cars observed by Caraoke as a function of time for both streets. The figure is marked with the green-yellow-red times for the traffic lights on each of the streets. The figure shows how a backlog of cars accumulates during a red-light and clears during a green light. Further,

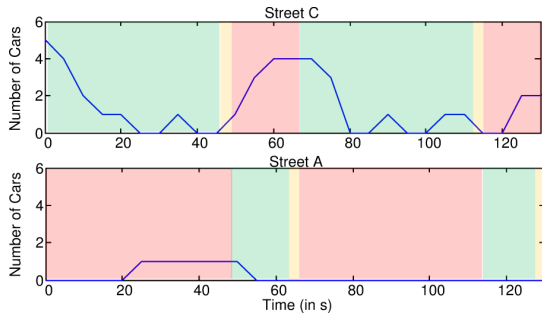


Figure 12—Traffic Monitoring at an Intersection. The figure shows the number of cars counted by Caraoke at an intersection over two periods of green-red lights.

road C is much busier than road A (10 times on average), but its green light is only 3 times longer than that of road A. This shows that Caraoke can help the city understand the relative congestion on various streets, and potentially adapt the timing of the traffic light to reduce car waiting time.

12.2 Localization Accuracy

In this experiment, we focus on localizing cars to parking spots. We ran our experiments on streets A and B, which have parking spots on one or both sides of the road, for a total of 36 spots. We use 4 poles, two deployed on street A and one at the intersection of A and B, and one on street B. We use two cars equipped with E-ZPass transponders, which we move between different parking spots to experiment with different configurations. There are other cars parked on the street, whose transponders collide with our two cars. We ran over 175 localization tests which span 35 configurations for the cars, and 5 runs per configuration which differ by the number and identity of the colliding transponders due to other parked cars and traffic dynamics on the streets. In our processing, we ignore the FFT spikes corresponding to other cars and focus on localizing our transponders using the method described in §6. To measure the ground truth we use a Bosch GLM50 laser distance measurement tool [17], which can measure distances to within a few millimeters. We then measure the ground truth angle using our knowledge of the transponder’s distance from the pole, the pole’s height, and the transponder’s elevation.¹⁴

Fig. 13 plots the error in the angle of arrival measurement computed by Caraoke. The errors are plotted as a function of the location of the parking spot with respect to the pole carrying the Caraoke reader. The bar graph shows the average and standard deviation of the measurements from all four poles. The figure shows that the average localization error is about 4 degrees. Interestingly the error is the largest at the two ends, i.e., when the car is only 1 spot away or 6 spots

¹⁴Measuring the ground truth for 3D angles is difficult. However, since the car is parked to the side of the road, the transponder, the pole and the two antennas are approximately in the same plane. Hence, knowing the distances we can measure the ground-truth angle with simple geometry rules. Note that this approximation means that our results are conservative –i.e., Caraoke’s actual errors could be slightly smaller.

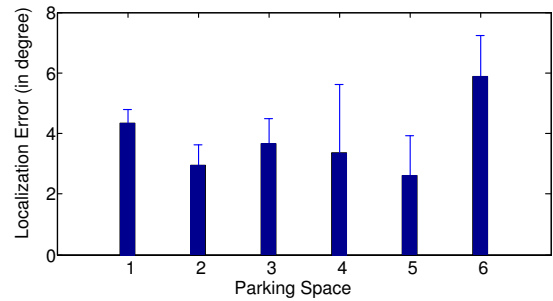


Figure 13—Transponder’s Localization Accuracy. Figure shows the accuracy of localization for cars parked in different parking spaces where the spot 1 is closest and spot 6 is the furthest spot to Caraoke pole. Caraoke enables enough accuracy to detect occupied versus available spots

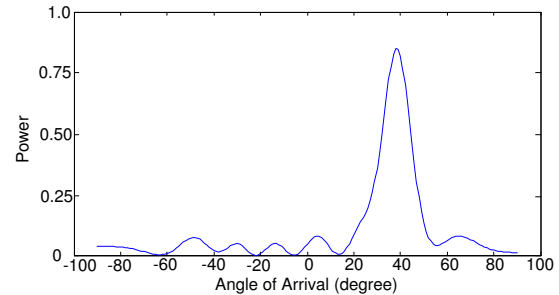


Figure 14—Multipath Profile. Caraoke readers have strong line-of-sight paths to the transponders. Therefore, multipath effects which occur in standard indoor environments are significantly weaker.

away from the pole. This is because the two antennas used for computing the AoA create a 60° angle with the plane of the road. As explained in §6, without this tilt the error in AoA for the furthest spot, i.e., spot 6, would be significantly larger than the closer spots. The results show that our decision of positioning the antennas at 60° angle with the street produces a relatively balanced error across spots.

One may be surprised that Caraoke’s AoA accuracy is high despite that it uses a simple two-antenna array. Large antenna arrays are typically needed in multipath scenarios to separate signals that travel along different paths from source to destination. While multipath effect has been a big challenge for indoor localization, it becomes less prominent in Caraoke’s design, primarily because Caraoke reader is mounted on a several meter high pole in an outdoor environment, and hence, has a prominent line-of-sight path between transmitter and receiver. To ascertain that this setting indeed has low multipath, we augmented Caraoke with an antenna attached to a rotating arm of radius 70cm. Like past work [37], we use this design to emulate a large antenna array (Synthetic Aperture Radar) and obtain the multipath profile of the signal coming from the car’s transponder. As the antenna rotates, we continuously measure the wireless channel of the transponder’s signal and then use the measured channels to reconstruct the multipath profile of the transponder’s signal using standard phased array processing algorithm and the MUSIC algorithm [60].

A representative multipath profile obtained using this setup is shown in Fig. 14. As expected, the multipath pro-

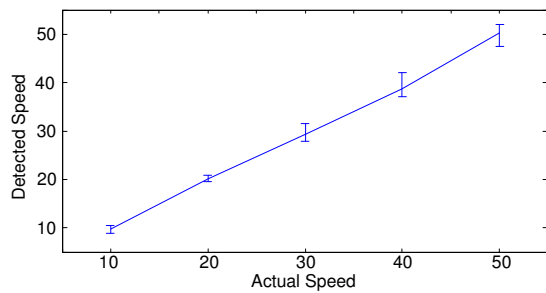


Figure 15—Speed Detection Accuracy. Figure shows the car speed detected by Caraoke versus its actual speed. Caraoke detected the speed to within 8% over a range of different speeds

file has one dominant peak. To confirm that this is indeed the case across experiments, we repeat the experiment for 100 runs across different times and locations and measure the relative power of the two peaks with the highest amplitude in the multipath profile (i.e., the highest to the second highest peak power). We observe that, on average, the strongest peak has an order of magnitude higher power (specifically, 27 times higher) as compared to the second strongest peak. This confirms our hypothesis, that for line-of-sight outdoor environments, multipath effects are significantly weaker than the line of sight peak and hence, do not interfere with accurate phase based localization.

12.3 Speed Estimation Accuracy

Next, we evaluate Caraoke’s ability to estimate car speed. For speeds below 40 mile/hour, we ran our experiments on street A and street D in Figs 10(a) and(b). We used two poles to localize the car and compute the speed as explained in 7. We locate the two poles 200 feet apart. Experiments with speeds higher than 40 mile/hour are performed in an empty lot. We perform 10 experiments at each speed, for a total of 50 experiments. We compare the speed detected by Caraoke with the speed reported by the car. Fig. 15 plots the estimated speed versus the actual speed. The plot shows both the average and the 90th percentile. The figure shows that Caraoke’s estimate of the speed is within 8% of the real value –i.e., the error is 1 to 4 mile/hour across the whole range of speeds.

12.4 Decoding Accuracy

We evaluate Caraoke’s ability to decode the ids of the transponders in the presence of wireless collisions. We place the Caraoke reader on a pole. The reader receives the colliding responses and decodes the ids of the transponders. Specifically, as described in §8, to decode a particular id, the reader combines the collisions after compensating for the channel and CFO of the desired transponder. The reader keeps combining collisions until the decoded id passes the checksum test. We use a maximum of 10 transponders whose owners agreed to the experiment. We run a total of 100 experiments with a different number of colliding transponders and different distances from the reader.

Fig. 16 shows the time taken to decode a transponder id for different numbers of colliding transponders. Recall that when decoding an id, Caraoke reader sends multiple queries,

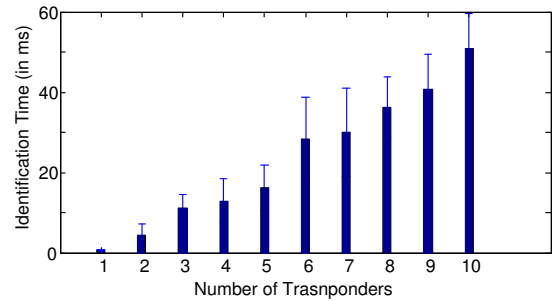


Figure 16—Identification Time versus Number of Transponders. As the number of transponders in Caraoke reader’s range increases, Caraoke requires longer sensing time to identify the tags.

and combines the resulting collisions to decode. Since the queries are separated by 1ms, the time axis also shows the number of combined collisions in order to decode. As can be seen, the required time increases as the total number of colliding transponders increases. This is because when decoding a transponder, other transponders in the collision act as noise. However, even when there are 10 colliding transponders, Caraoke can still decode the transponder of interest within 50ms, on average.

Note that 50ms is also the time to decode all 10 transponders since one does not need to collect new collisions for individual transponders. One only needs to compensate for the CFO and channel of each of the transponders differently. Since the processing time of each query is negligible in comparison to the time it takes to transmit and receive, decoding all colliding transponders takes as much total time as decoding one transponder.

12.5 Caraoke Reader’s Power Consumption

To profile Caraoke reader for power consumption, we removed the solar panel and the battery and used the USB port to power the device. We connected the USB to In-line Voltage and Current Meter [13], which measures the voltage and current drawn by the board. The power measurements reveal that Caraoke reader consumes 900mW in active mode and consumes only 69 μ W in sleep mode.¹⁵ Since the solar panel delivers 500mW in the sun, Caraoke reader would not be able to run continuously in the active mode. However, as explained in §10, due to duty cycling, the average power consumption of the board is much lower as the duration of the active mode is less than 10ms. Thus, if Caraoke reader takes one measurement every second, it would consume an average power of 9mW, which is $\sim 56\times$ lower than what it can harvest from its solar panel. Hence, the energy harvested from solar during 3 hours can be stored in a rechargeable

¹⁵These numbers exclude the modem module. LTE and Wi-Fi modems consumes 1-2 W and 100s of mW, respectively, while transmitting at Mbps data rates. A Caraoke reader needs to transmit only a few kbits to convey the results of processing one query (i.e., the channels and CFOs) to a backend. Furthermore, it can batch the results of multiple queries together. Hence, it can use the modem for tens of milliseconds then put it to sleep for a minute or so. By duty cycling the modem, one can bring down its average power consumption to mW or hundreds of μ W.

battery and run the device for a week regardless of weather condition. Finally, note that this operation time can be further increased by using a larger solar panel or increasing the sleep time.

13. CONCLUSION

This paper presents Caraoke a system that can count, localize, and measure the speed of cars using the RF signal from their e-toll transponders. Caraoke readers are small, low-cost and low-power, and hence can be easily deployed on street lamps to allow cities to deliver smart services, e.g., smart parking, traffic monitoring and speed detection, all using one infrastructure. While we focused mainly on tracking vehicles, once such infrastructure exists, the city may use it for additional services, like locating first responders, tracking the delivery of goods, and enabling people to pay for mobile services.

Acknowledgments: We thank Haitham Hassanieh, Fadel Adib, Hariharan Rahul, the reviewers and our shepherd, Lin Zhong for their insightful comments. This research is supported by NSF. We thank the members of the MIT Center for Wireless Networks and Mobile Computing, including Amazon.com, Cisco, Google, Intel, MediaTek, Microsoft, and Telefonica, for their interest and support.

14. REFERENCES

- [1] E-ZPass statistics. <http://e-zpassag.com/about-us/statistics>.
- [2] E-ZPass taking toll on turnpike collectors. http://www.meadvilletribune.com/news/local_news/article_735f7ca1-82f9-5340-8940-29608aa499d5.html.
- [3] NTP org: NTP over cellular. http://support.ntp.org/bin/view/Support/ConfiguringNTP#Section_6.15.
- [4] PA turnpike to make E-ZPass mandatory. <http://www.bizjournals.com/philadelphia/news/2013/08/29/pa-turnpike-to-make-e-zpass-mandatory.html>.
- [5] Paying with E-ZPass. <http://www.panynj.gov/airports/lga-paying-with-e-zpass.html>.
- [6] The problems with police radar. <http://www.ibiblio.org/rdu/a-btrust.html>.
- [7] Solar powered RFID system. <http://www.gizmag.com/savi-technology-solar-rfid/8373/>.
- [8] Speed limits in the united states. http://en.wikipedia.org/wiki/Speed_limits_in_the_United_States.
- [9] Tunpike tolls to be all-electronic by 2018. <http://paindependent.com/2013/08/no-e-zpass-heres-a-bill-electronic-tolling-planned-for-pa-turnpike>.
- [10] Urban design division/planning and development review department. http://austintexas.gov/sites/default/files/files/Planning/Urban_Design/great_street_site_overview_0512.pdf.
- [11] O. Abari, E. Hamed, H. Hassanieh, A. Agarwal, D. Katabi, A. P. Chandrakasan, and V. Stojanovic. A 0.75-million-point fourier-transform chip for frequency-sparse signals. In *IEEE ISSCC*, 2014.
- [12] O. Abari, H. Rahul, D. Katabi, and M. Pant. Airshare: Distributed coherent transmission made seamless. In *IEEE INFOCOM*, 2015.
- [13] Adafruit. USB Voltage and Current Meter. <http://www.adafruit.com/product/1852>.
- [14] A. Amini, K. Kung, C. Kang, S. Sobolevsky, and C. Ratti. The impact of social segregation on human mobility in developing and industrialized regions. *EPJ Data Science*, 2014.
- [15] Analog Device. ADC 7356. http://www.analog.com/static/imported-files/data_sheets/AD7356.pdf.
- [16] J. A. Bonneson and M. Abbas. Intersection video detection manual. Technical report, Texas Transportation Institute, Texas A & M University System, 2002.
- [17] Bosch Tools. Glm 50 laser distance measurer. <http://www.boschtools.com/>.
- [18] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *ACM MobiCom*, 2008.
- [19] CBS Local. New app allows Boston drivers to pay for parking on smartphone. <http://boston.cbslocal.com/2015/01/14/new-app-allows-boston-drivers-to-pay-for-parking-on-smartphone/>.
- [20] W. M. da Silva, A. Alvaro, G. H. R. P. Tomas, R. A. Afonso, K. L. Dias, and V. C. Garcia. Smart cities software architectures: A survey. In *ACM SAC*, 2013.
- [21] S. Eichler. Performance evaluation of the ieee 802.11p wave communication standard. In *Vehicular Technology Conference*, 2007.
- [22] EPCglobal Inc. Electronic Product Code (EPC): An overview. http://www.gs1.org/docs/epcglobal/an_overview_of_EPC.pdf.
- [23] EPCglobal Inc. EPCglobal class 1 generation 2 v. 1.2.0. <http://www.gs1.org/gsm/kc/epcglobal/uhfc1g2>.
- [24] Escort Radar. The truth about speed enforcement. https://www.escortradar.com/pdf/radar_report.pdf.
- [25] European Commission. Smart Cities: Digital agenda for Europe. <http://ec.europa.eu/digital-agenda/en/smart-cities>.
- [26] Z. Farid, R. Nordin, and M. Ismail. Recent advances in wireless indoor localization techniques and systems. *Journal of Computer Networks and Communications*, 2013.
- [27] Federal Communications Commission. Title 47, code for federal regulations, part 15.
- [28] FOX News. Company lets drive-thru customers pay for fast food with E-Zpass. <http://www.foxnews.com/leisure/2013/12/16/fast-food-drive-thrus-get-faster-company-lets-customers-pay-with-e-zpass/>.
- [29] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: enabling phased array signal processing on commodity WiFi access points. In *MobiCom*, 2014.
- [30] GS1. Electronic Product Code Implementation. <http://www.gs1.org/epcglobal/implementation>.
- [31] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse fourier transform. In *STOC*, 2012.
- [32] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse FFT. In *SODA*, 2012.
- [33] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi. Bigband: Ghz-wide sensing and decoding on commodity radios. In *IEEE INFOCOM*, 2014.
- [34] P. Hu, P. Zhang, and D. Ganesan. Fully asymmetric backscatter communication. In *ACM SIGCOMM*, 2015.
- [35] IEEE. IEEE Smart Cities. <http://smartcities.ieee.org/>.
- [36] Kapsch TrafficCom. Active TDM LMS specifications. <http://tdm.kapschtraffic.com/>.
- [37] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li. LTE radio analytics made easy and accessible. In *ACM SIGCOMM*, 2014.
- [38] KY3. Springfield's high tech cameras count cars to ease traffic congestion. http://articles.ky3.com/2012-06-20/springfield-traffic_32340154.
- [39] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2007.
- [40] Y. Liu, J. Bi, and J. Yang. Research on vehicular ad hoc networks. In *CCDC*, 2009.
- [41] Maxim Integrated. MAX2117. <http://datasheets.maximintegrated.com/en/ds/MAX2117.pdf>.

- [42] McKinsey & Company. The smart-city solution. http://www.mckinsey.com/insights/public_sector/the_smart-city_solution.
- [43] J. C. Medina, M. Chitturi, R. F. Benekohal, and T. R. Board. Illumination and wind effects on video detection performance at signalized intersections. *Urbana*, 2008.
- [44] L. Mojela and M. Booyesen. On the use of WiMAX and Wi-Fi to provide in-vehicle connectivity and media distribution. In *ICIT*, 2013.
- [45] C. D. Murta, P. R. Torres Jr, and P. Mohapatra. Characterizing quality of time and topology in a time synchronization network. In *GLOBECOM*, 2006.
- [46] New Jersey Turnpike Authority. Travel resources: E-ZPass information. <http://www.state.nj.us/turnpike/ez-pass.html>.
- [47] OSEPP. Monocrystalline Solar cell. <http://osepp.com/products/solar-cells-arduino-compatible/>.
- [48] H. Rahul, H. Hassanieh, and D. Katabi. SourceSync: A distributed wireless architecture for exploiting sender diversity. In *ACM SIGCOMM*, 2010.
- [49] H. Rahul, S. S. Kumar, and D. Katabi. MegaMIMO: Scaling wireless capacity with user demand. In *ACM SIGCOMM*, 2012.
- [50] Sequans Communications. Chipset solution for category 1 LTE devices for the internet of things. <http://www.sequans.com/products-solutions/streamlitel/calliope-lte-platform/>.
- [51] Streetline. Streetline Inc. <http://www.streetline.com/>.
- [52] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang. Fine grained channel access in wireless LAN. In *SIGCOMM*, 2010.
- [53] Texas Instruments. Energy harvesting. http://www.ti.com/corp/docs/landing/cc430/graphics/slyy018_20081031.pdf.
- [54] Z. Tian, M. Kyte, and H. Liu. Vehicle tracking and speed measurement at intersections using video detection systems. *ITE Journal*, 2009.
- [55] Transportation Corridor Agencies. FasTrak on new I-10 and I-110 ExpressLanes. <https://www.thetollroads.com/communications/switchable-transponders-110.php>.
- [56] Transportation Corridor Agencies. What is FasTrak? how do i use it to pay tolls? <https://www.thetollroads.com/ontheroads/commonquestions/fastrak.php>.
- [57] R. J. Vullers, R. Schaijk, H. J. Visser, J. Penders, and C. V. Hoof. Energy harvesting for autonomous wireless sensor networks. *IEEE Solid-State Circuits Magazine*, 2010.
- [58] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. In *ACM SIGCOMM*, 2012.
- [59] J. Wang and D. Katabi. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. In *ACM SIGCOMM*, 2013.
- [60] J. Xiong and K. Jamieson. ArrayTrack: A fine-grained indoor location system. In *NSDI*, 2013.
- [61] Z. Xiong, H. Sheng, W. Rong, and D. Cooper. Intelligent transportation systems for smart cities: a progress review. *Science China Information Sciences*, 2012.
- [62] S. Xu, P. Guo, B. Xu, and H. Zhou. QoS evaluation of VANET routing protocols. *JNW*, 2013.
- [63] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *ACM MobiCom*, 2014.
- [64] T. Zinchenko, H. Tchouankem, L. Wolf, and A. Leschke. Reliability analysis of vehicle-to-vehicle applications based on real world measurements. In *VANET*, 2013.