CrowdPark: A Crowdsourcing-based Parking Reservation System for Mobile Phones

Tingxin Yan§, Baik Hoh†, Deepak Ganesan§, Kenneth Tracton†, Toch Iwuchukwu†, Juong-Sik Lee†
§Dept. of Computer Science, University of Massachusetts, Amherst MA 01003
†Nokia Research Center, Palo Alto CA 94304
{yan, dganesan}@cs.umass.edu§, {baik.hoh, ken.tracton, toch.iwuchukwu, juong-sik.lee}@nokia.com†

ABSTRACT

Parking in crowded urban areas is a precious resource that impacts driver stress levels, daily productivity, and the environment. A reservation system that enables individuals to buy parking spots prior to leaving their home would significantly ease these concerns. However, designing an infrastructure for guaranteed parking requires extensive sensor deployment and manpower, which is expensive and time-consuming proposition.

In this paper, we present CrowdPark, a crowdsourcing platform that enables users to "loosely reserve" parking spots. Unlike traditional reservation platforms where sellers are usually the owners of resources, CrowdPark achieves parking reservation by crowdsourcing information about when parking resources will be available, and using this availability information to help other users find parking spots. The design of such a crowdsourcing-based parking reservation system presents several challenges including incentive design, robustness to malicious users, and handling the spatial and temporal uncertainty due to real-world vagaries. We present novel solutions to address these challenges that combine protocol design, game-theoretic and cost-benefit analysis, sensor data processing techniques, and navigation-based tools. With a combination of simulation and real-world experiments, we show that CrowdPark can 1) effectively incentivize user participation and detect malicious users with accuracy of over 95%, and 2) handle over 95% of spatial uncertainty and achieve over 90% successful parking reservation with a few minute long waiting time.

1. INTRODUCTION

Parking in crowded urban areas is a precious resource and drivers spend substantial amounts of time locating empty parking spots. Metropolitan cities such as San Francisco and New York City, in particular, have a pressing problem due to limited parking in downtown areas, both for street as well as garage parking [28]. Making matters worse, information about parking availability is typically unavailable to drivers. Studies by the US Department of Transportation have reported that parking patrons "often do not know where the best parking locations are", and "most importantly, whether a parking place will be available when they arrive" [23].

The chasm between demand versus supply of parking spots causes a spectrum of environmental, health and safety issues. Drivers keep vehicles on the road in the process of circling (usually slowly) the areas where they want to locate parking. This leads to lengthy queues of vehicles that can block several streets. In addition, the increased acceleration, deceleration, and braking behavior while circling has significant impact on automobile emissions [2]. Parking issues also adds to driver stress levels, and has been reported to increase road rage and accidents [6].

These concerns have led to significant efforts to design online real-time parking information systems that provide up-to-date information about parking availability. For example, SFPark is a pilot project to monitor real-time parking availability in San Francisco by deploying a massive network of sensors [27]. While such parking information systems can help direct drivers to available parking locations, they are in their pilot stages and face daunting scaling and budgetary challenges given the vast volume of street parking in U.S. cities. Continuous monitoring of street parking requires installation of occupancy sensors on hundreds of thousands of parking spots or parking meters, and a vast wireless infrastructure to obtain and transmit sensing data in a reliable manner.

The difficulties in deploying a continuous-sensing based parking infrastructure has led to increased interest in the use of crowdsourcing using mobile phones. Several mobile applications such as Google's OpenSpot [14] and PrimoSpot [3] were recently released, with the intent of using the general public to locate empty parking spots. Compared with infrastructure-based approaches such as SFPark [27], crowdsourcing-based approaches offer higher agility, lower cost, and larger coverage since it utilizes the vast number of mobile phone users. Recent research has also shown that using distance sensors attached to taxis [21] for obtaining parking availability information is promising, although this is a more expensive and less scalable approach than using mobile phones. A fundamental limitation in all these systems is that they are designed to share parking availability information and do not provide any information about when a parking space is taken. Parking availability information, by itself, is not particularly useful since empty spots in crowded areas are quickly consumed. Also, these systems do not address several fundamental challenges including incentives for users to provide information, or mechanisms to verify the validity of this information.

In this paper, we address this limitation of crowdsourced parking systems by designing CrowdPark, a crowdsourcing platform that enables users to "loosely reserve" parking spots. CrowdPark is based on the idea that an individual who is currently parked can provide advance notification about when they plan to leave, and this information may be sold to a buyer who is willing to pay for reserving the parking spot. The buyer arrives at the reserved parking spot close to the leaving time, and can occupy the spot when the seller leaves. Since individuals transact parking availability information as oppose to concrete parking spots, CrowdPark presents a loose reservation model in contrast to a guaranteed service such as valet parking. However, CrowdPark requires no additional infrastructure or man-power, making the system considerably less expensive that infrastructure-based approaches.

Although the vision of loose parking reservations is intriguing, a practical realization of such a crowdsourcing-based parking reservation system presents several hurdles. First, such a system should encourage participation, both by providing incentives for first-time sellers to contribute their leaving time information, and by encouraging buyers to re-sell the spots that they successfully reserve. Second, the incentive mechanism should take into account the "utility" of a transaction — bonuses should be provided when a transaction leads to successful parking, and refunds when unsuccessful. Third, there is a need to handle malicious users whose goal is solely to maximize their monetary gain, either by masquerading as fake sellers or by denying successful reservations to obtain a refund. Fourth, there is a need to accurately identify a specific parking spot since location error can lead to low success rates of reservation. Fifth, there is a need to handle uncertainty in seller and buyer behavior — a seller may not leave at the time indicated and the buyer may not arrive at that time, leading to missed opportunities for a successful reservation.

CrowdPark addresses these challenges using a novel combination of incentive designs, sensor data processing techniques, and navigation-based tools. Our incentive protocol is designed to create a vibrant marketplace for parking information, and incentivizes participation, provides utility-based bonuses, offers refunds for failed reservations. We analytically show that by choosing payment and incentive parameters carefully, we can discourage malicious buyers and ensure that the service provider always makes a profit. However, not all challenges can be addressed solely by adjusting payment parameters, and we turn to phone-based sensing and user interaction to address these. We develop 1) techniques that use activity recognition and image processing to detect malicious sellers, and buyer-seller cooperation techniques assisted by route navigation tools to address uncertainty in location and time.

The CrowdPark system is implemented with a central server and a mobile client on HTML-5 enabled smartphones. We show that:

- Our game-theoretical design and cost-benefit analysis ensures that rational users act honestly and that CrowdPark is profitable for the service provider.
- Our sensing-based approach can detect malicious users with close to 100% accuracy when they are pedestrians and over 95% when they are motorists.
- Our positioning scheme can identify the precise locations of reserved parking spots up to 95% of the time in the San Francisco downtown area.
- CrowdPark can detect late arrival of buyers with accuracy of a few minutes in San Francisco downtown area.
- With cooperation from sellers, CrowdPark can achieve over 90% successful reservation rate.

2. CROWDPARK RESERVATIONS

CrowdPark provides a platform where users can trade parking availability information. There are two types of users in CrowdPark system: sellers and buyers. Sellers are the drivers who occupy parking spots, and sell their "when-to-leave" (henceforth referred to as WTL) information. Buyers are the drivers who buy the WTL information to reserve parking spots. CrowdPark uses virtual credits as the incentive for the exchange of WTL information between sellers and buyers — several providers including Facebook support virtual credits, which can be exchanged for monetary rewards or products in a store [1]. This section describes the basic reservation protocol underlying CrowdPark, and discusses several open challenges that we address in the coming sections.

2.1 Reservation Protocol

A WTL transaction, shown in Figure 1, comprises three major steps: 1) a seller submits a WTL message to CrowdPark, 2) a buyer buys a WTL message to reserve a parking spot, and 3) the buyer drives to the reserved parking spot, and sends back the reservation result. We explain these three steps in more detail as follows.

Seller submits WTL to CrowdPark . In order to incentivize users to share WTL information, CrowdPark pays sellers two types of rewards - a fixed reward of D points and a bonus of X points. The fixed reward is granted immediately after a WTL message is accepted by CrowdPark. This incentive is provided irrespective of whether the spot is purchased by a buyer. Such a participation incentive can help bootstrap our system, and ensure a steady stream of available slot information to make the marketplace attractive to buyers. This fixed

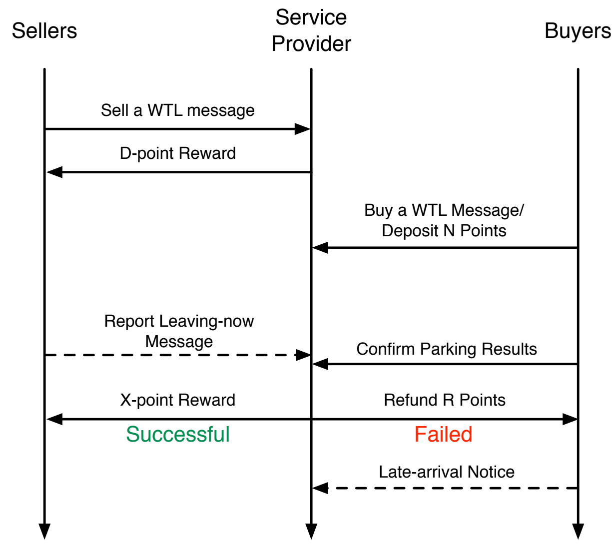


Figure 1: The transaction process of a WTL message. The messages marked with dotted lines are not mandatory, and will be explained in later sections.

reward is expected to be a small amount, say a quarter, although this may be adjusted based on the importance of the location and time of the sale — for example, the reward for downtown San Francisco may be higher than suburban areas, and can vary over time depending on parking need. The bonus reward is granted right after the parking reservation is confirmed a success *i.e.* after a buyer has successfully parked at the seller's location. This amount is expected to be significantly larger than the fixed reward, say one or a few dollars depending on the location and time.

A WTL message contains location and time information of the parking spot, to allow a buyer to accurately locate a spot and arrive at the spot in a timely manner. Location information is specified by a combination of inputs: 1) GPS location at the parking spot, 2) identifier of the parking spot, and 3) an identification of the seller's vehicle such as the color, make, and license plate number of the seller's car. Note that the sellers vehicle identification is collected only once upon registering with CrowdPark and does not have to be re-sent each time. §5.1 describes how these inputs are obtained and combined to determine location of the spot. The time information is specified by the expected leaving time of the seller.

Buyers buy WTL to reserve parking spots . A buyer who wishes to purchase a parking spot provides a destination and arrival time to CrowdPark, and receives a list of matching WTL messages. The returned WTL messages are first presented to buyers as an "overview" mode, where the precise information in a WTL is obfuscated. For example, the actual time is replaced by a

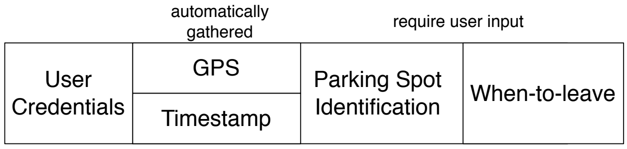


Figure 2: Contents of a WTL message sent by a seller. Parking spot identification and "when-to-leave" field are provided by seller. CrowdPark combines this with seller's vehicle identification information to form a full WTL message for reservation.

time window of 10 minutes, and the location is replaced by the distance to buyers' destination. The "overview" mode allows buyers to identify the appropriate WTL messages while not to revealing the WTL information in its entirety before buyers pay for it. After a buyer pays for a WTL message, the actual content of the WTL, including the location and leaving time information, is visible to the buyer. The reservation is complete at this point. The payment from buyers, N points per WTL message, is deposited to CrowdPark.

Buyers confirm reservation results The final step is a confirmation from the buyer about the success or failure of the parking reservation. If a reservation is successful, the seller receives a bonus of X; if it fails, a refund R is returned to the buyer. Note that refund R may be different from the deposit paid by the buyer, also the base reward D depends on the CrowdPark dynamics as well as the desired profit margin for the service provider. We discuss this in detail in § 3.

2.2 Challenges

The reservation protocol underlying CrowdPark leaves several unanswered questions. Since monetary incentives are involved, one class of problems that CrowdPark needs to solve is how to deal with malicious sellers or buyers who may be willing to lie to maximize their rewards. A second class of problems involves dealing with real-world vagaries such as location error, early departure of the seller, or late arrival of the buyer. More specifically, there are four key challenges that we address in the rest of this paper:

- *How to ensure that buyers are honest?* A malicious buyer can buy a reservation and park successfully but deny the fact to obtain a refund. In § 3.1, we introduce a game-theoretical design that ensures rational buyers maximize their gain by confirming the truth.
- *How to ensure profitability for the service provider?* The parameters of the reservation protocol influence the margins for the parking service provider. In §3.2, we provide a cost-benefit analysis from

Fixed incentive for seller	D
Seller bonus for successful reservation	X
Buyer deposit for reservation	N
Buyer refund for unsuccessful reservation	R
Probability of successful reservation	p
Probability of unsuccessful reservation	q
Alternate revenue (e.g. ads)	C

Table 1: CrowdPark Parameters.

the provider perspective and describe how it can be used to determine protocol parameters.

- *How to make CrowdPark robust against malicious sellers?* Malicious sellers might provide fake WTL information to receive base reward. In section 4, we introduce two sensing based schemes, ActCheck and SpotCheck, to detect malicious sellers.
- *How to precisely localize the parking spot?* A parking reservation system relies on the ability to precisely pinpoint the location of the spot being reserved. But GPS location is often inaccurate, particularly in downtown areas with tall buildings. In §5.1, we describe how a combination of GPS-based processing, parking meter identifiers, and car identification information can enable precise localization.
- *How to handle uncertainty in buyer arrival and seller departure?* Despite the best intentions, it is difficult to precisely arrive at the parking spot at the WTL time due to traffic vagaries, and precisely leave at that time due to scheduling difficulties. In §5.2, we discuss predictive techniques to inform the buyer/seller of delays and enable better synchronization, and utilize cooperation of sellers and buyers to improve reservation success rate.

3. SETTING CROWDPARK PARAMETERS

How to fully incentivize users to act honestly in CrowdPark while meet a budget for service providers? In this section, we provide a novel game-theoretical design and cost-benefit analysis, where we achieve this goal by carefully setting the different reward and payment variables such as D, R, and X in the system.

3.1 Incentivizing Honest Buyers

The first challenge that we address is dishonest buyers. In the protocol described earlier, a buyer can deny that a parking was successful even if it was indeed successful, and thereby receive a refund. Buyer dishonest also has the negative consequence that the seller is not being given a bonus, which can reduce participation. We now present a game theoretical design of the parking reservation rules which ensures that buyers maximize their gain by telling the truth. With such a design, rational buyers would choose to be honest in our system.

	Honest	Dishonest
Successful	$D + pX$	R
Unsuccessful	R	D

Table 2: The gain for buyers by being honest and dishonest when reservations are successful and unsuccessful.

The key idea behind our approach is that a buyer who successfully parks at a reserved spot can re-sell that spot through the CrowdPark system if they tell the truth. If they deny a successful parking, obviously they cannot re-sell this spot. We use this idea to set reward parameters to ensure that the average gain of re-selling is higher than the gain by lying and receiving a refund. The approach has two benefits: a) it incentivizes honest reporting from buyers, and b) it encourages buyers to keep re-selling a spot thereby ensuring that a steady pool of spots are available on CrowdPark. We now describe the approach more formally.

As mentioned in the previous section, the gain of selling a WTL message contains two parts: a constant reward D, and a bonus reward X. If the probability of selling a WTL and confirmed successfully is p, the reward for selling a WTL message is: $D + pX$. To ensure that a buyer is better off being truthful, we only need to ensure that the average reward of reselling a reserved parking spot is higher than R.

Table 3.1 shows the gain for a buyer if they tell the truth vs lie, and when parking was successful vs unsuccessful. To ensure buyers maximize their gain by being honest, we need the following constraint to hold:

$$D + pX \geq R \geq D \tag{1}$$

$$\text{or,} \quad X \geq \frac{1}{p}(R - D) \tag{2}$$

The above analysis assumes that we know p, the probability that a WTL is sold and confirmed successful. In a live system, p is simply a measured system parameter, and the current measured value of p can be used in the inequality.

While Equation 2 provides a lower bound for X, the bonus has to be upper bounded as well, since a service provider has profit considerations.

3.2 Service Provider Cost-Benefit Analysis

Under what conditions is CrowdPark a profitable business enterprise? As we will show, the analysis for this question helps further narrow down the region of the values for bonus X and refund R.

Let's consider a single WTL transaction. For each WTL message, the *cost for the service provider* is as follows: 1) reward to the seller is $D + pX$, where p is the probability that a WTL is sold and confirmed successful, and 2) refund to the buyer is qR, where q is the probability that a WTL is sold and confirmed unsuccessful. The *benefit for the service provider* is the income that can be earned from each WTL, which is $(p + q)N$.

Mobile businesses typically have multiple sources of revenue such as targeted advertisements (e.g. for parking garages), so they may not be reliant solely on income from parking reservations. Let us assume that service provide has a per-WTL revenue of C from these alternate sources. Our goal is to ensure that service provider can break even. In other words, we have the following constraint:

$$C + (p + q)N \geq D + pX + qR$$
$$\text{or,} \;\; X \leq \frac{1}{p}(C + (p + q)N - D - qR) \quad (3)$$

Together with (2), we have upper and lower bounds for bonus X as follows:

$$\frac{1}{p}(R - D) \leq X \leq \frac{1}{p}(C + (p + q)N - D - qR) \quad (4)$$

The above inequality implies that:

$$R - D \leq C + (p + q)N - D - qR$$
$$\text{or,} \;\; R \leq \frac{1}{(1 + q)}(C + (p + q)N)$$

Together with (1), we can derive the constraints for refund R as follows:

$$D \leq R \leq \frac{1}{1 + q}(C + (p + q)N)$$
$$\text{or,} \;\; D \leq \frac{1}{1 + q}((p + q)N + C) \quad (5)$$

This is the constraint for base reward D and WTL price N given a budget C from service provider.

Example We now instantiate the above analysis with a simple example. Let $C = 0$, which means that there is no additional source of income, and CrowdPark needs to be profitable on its own. Then, we can derive the relation between D and N based on (5) as follows:

$$D \leq \frac{1}{1 + q}((p + q)N)$$
$$\text{or,} \;\; \frac{D}{N} \leq \frac{p + q}{1 + q} \quad (6)$$

This is the condition that CrowdPark is profitable. We can change the ratio of D and N based on the reservation probability, p that we learn over time. For example, assume that $N = \$2$, and that $p = q = 0.1$. We can derive other system parameters as follows:

► From (6), we have $D \leq \$0.36$.

► From (1), we have $\$0.20 \leq R \leq \$0.36$.

► From (4), we have $\$1.5 \leq X \leq \$1.65$.

Thus, we have derived system parameters D, R, and X that encourage honesty from buyers and ensure that the service provider is profitable from CrowdPark.

4. DETECT MALICIOUS SELLERS

While reservation parameters can be chosen to encourage honesty from buyers, it cannot address the issue of dishonest sellers. A dishonest seller gets a fixed reward, D, whether or not a reservation is successful and can therefore submit a large amount of incorrect parking information to the system. Such dishonest sellers can greatly degrade the experience for the buyer and adversely impact the viability of CrowdPark.

In this section, we describe three approaches to detect malicious sellers, and how these approaches can be used in conjunction to design a robust detection system. Throughout this discussion, we assume that users can be compromised but the phones are trustworthy. Thus a malicious user cannot manipulate GPS coordinates and timestamp intentionally. We also assume that malicious users cannot have multiple accounts, which is reasonable since it would require a separate vehicle, credit card, and phone for each account.

Baseline schemes A few simple baseline mechanisms can be used as a first-level filter for preventing malicious sellers. First, the service provider can pre-determine parking hotspots that are considered valuable to users and reject WTLs originating from other areas. Second, the service provider can rate limit the number of WTL messages per day. If a seller generates more than the limit, the service provider can place the seller in a blacklist. The advantage of these approaches is that both are easy to implement at the server-side and require no additional information from the phone. However, these techniques cannot completely prevent malicious sellers — hotspot areas can be easily guessed by a malicious seller, and rate limits typically need to be set conservatively and a malicious seller could sell several spots per day once they know the limit.

SpotCheck using geo-tagged licence plate A simple yet effective measure to detect whether a seller is actually parked at the claimed location is to require the seller to provide a geo-tagged photo of the vehicle's license plate using their phone. The license plate number can be accurately extracted from the photo using automatic techniques for Automatic Number Plate Recognition (ANPR [20]), or if the accuracy of these approaches is low, can be performed at low cost using human computation on the Amazon Mechanical Turk. The geo-tag associated with the photo can be cross-checked against the claimed location in the WTL message, and the license plate number in the photo can be matched against the information provided by the seller during registration. The benefit of this approach is that it provides an in-scene proof. However, it requires the seller to take a photo and transmit it to the server, which is time-consuming for the seller and energy-intensive for the phone. Another problem is that image quality can vary significantly, due to the reasons such as lighting, exposure. For example, if a vehicle is parked at night, it may be harder to get a clear image of the license plate due to lighting conditions. Also, a malicious seller could take an in-scene photo of not the plate itself, but a previously obtained photo of the license plate. It also requires image processing techniques have the ability to detect duplicated images. While these image processing is challenging, we could utilize crowdsourcing based human recognition to provide highly accurate results within a few minutes delay [32].

ActCheck using activity recognition on the phone A third approach is to use activity recognition techniques on the phone to detect if the activities performed by the seller is consistent with expectations. We expect that a normal seller be detected walking between the WTL submission and the leaving time, and within a moving vehicle after the leaving time. Activity recognition using phones is a well-studied research area, and several state-of-the-art algorithms[5, 18, 19, 30, 31] exist for accurately detecting activities such as walking, driving, biking, etc in a power-efficient manner. For example, the system that we use, Jigsaw[19] , uses an accelerometer to detect high frequency vibration and speed to infer user activity. By primarily using the accelerometer rather than the GPS, Jigsaw can reduce battery consumption.

ActCheck requires one additional message from the seller, a "Leaving Now" (LN) message that informs the system that they are leaving at a particular time. While the leaving time can be inferred from the activity, it provides more explicit information to the system.

ActCheck consumes less energy than SpotCheck since it requires limited communication, requires no explicit work from the seller, and uses lightweight and power-efficient sensors. However, it requires that the application be running continuously on the sellers phone. It also assumes a specific activity model for the seller, which can both lead to attacks from a malicious seller and lead to false positives when a legitimate seller exhibits behavior that is detected to be abnormal. For example, a malicious seller can deceive the system by slowly driving after WTL submissions to act as a vehicle owner who parked and walked out to the destination. Our experience with the Jigsaw system indicates that such deception is very hard to achieve in practice — malicious sellers need to perform a perfect combination of extremely slow driving (less than 10 mph), shaking the phone vigorously while driving to give the impression of walking, and following specific driving loops, in-order to deceive the system. Perhaps more problematically, ActCheck might trigger a false positive for legitimate activity that does not follow the normal behavior — for example, a user might send a WTL while arriving via a subway train to the parking lot. We do not consider all such cases in this paper, but as shown in [30], many of them can be detected by combining GPS information, bus/train schedules, and activity information.

Combining above approaches: While there are several approaches, each of them has pros and cons. The baseline schemes are the easiest to implement at the server-side, and provide a first-order filter against malicious sellers. When the baseline schemes detect a pattern, for example, a seller whose WTL information is repeatedly unsuccessful, or never bought, ActCheck is triggered. ActCheck provides a second-order filter to detect if the activity pattern of the seller follows expected behavior. If ActCheck triggers alarms for a seller, we trigger SpotCheck for future sales from the same seller as a more reliable mechanism for detecting malicious behavior.

5. DEALING WITH UNCERTAINTY

In addition to malicious buyers and sellers, a fundamental set of real-world challenges that need to be addressed in CrowdPark is location and time uncertainty. Precise location estimation of a parking spot is critical to the success of CrowdPark. GPS location, by itself, is insufficient since GPS error can be large in urban areas. Time uncertainty is also a crucial challenge — sellers may not leave at the precise time stated in the WTL message, and buyers may not arrive at this time either. In the rest of this section, we present our solutions to these challenges.

5.1 Handling Spatial Uncertainty

CrowdPark uses a mixture of GPS localization and physical identification to accurately locate parking spots. First, we use GPS to achieve street segment level positioning accuracy – identifying which street or which parking garage corresponds to the spot. Second, we use physical identification of parking spots such as a parking meter identifier, the level/section number in a garage, or the color/make/license number of the seller's car to identify the parking location. We discuss these techniques in more detail below:

Street-Segment Matching: The first challenge in handling spatial uncertainty is finding the correct street segment. A single GPS reading is not enough since GPS readings can change significantly even with small perturbations in urban areas. Thus, even waiting for a longer duration does not ensure that GPS converges to the right location. In CrowdPark, we use a simple Street-Segment matching algorithm to reduce the error of GPS readings, and to find the correct street segment with high probability.

Figure 3: Parking meter ID. Left one in San Francisco and right one in Amherst, MA

CrowdPark reads GPS samples for a few minutes after sellers register their parking spot with the server. For each sample, we match it to the street segment which has the minimum distance to the sample and has valid street parking. For all the street segments that have one or more samples matched, we rank them by the number of samples matched. We then return the top ranked street segments which in total contains over 90% of samples to sellers, and let sellers choose the correct street segment from the results.

Street Parking Spot Identification: The street-segment matching algorithm has two problems: 1) it provides only coarse-grained information about the parking spot and a buyer needs to search the entire street segment to identify the specific seller spot, and 2) the algorithm cannot provide 100% accuracy due to GPS inaccuracy, particularly in urban areas.

To address this issue, we use physical identification to locate the parking spot. The best physical identifier of a parking spot that we found is the parking identification meter number that is available on all parking meters. (An example is shown in Figure3). This number on the parking meter does not appear to be unique across the U.S. but is unique in a local scope (such as a district). The number of digits on the meter also vary — there are ten digits in San Francisco and three digits in Amherst. Since we already obtain street-segment level information from GPS, we only require that they are locally unique within the small region. Thus, a seller only needs to enter the last three digits of the parking identifier to precisely pinpoint the location of the spot.

The discussion above assumes that we have access to a database of parking meter identification and their GPS coordinates. We have not been able to find an open database with this mapping; however, even if such a database is not available from official sources, obtaining this information is a one-time overhead. In fact, this data collection can even be crowdsourced by recruiting sellers to contribute to the database by providing input of additional information associated with the meter, such as the address of the closest building, or simply mark it on a Google Map street view interface.

5.2 Handling Temporal Uncertainty

The "when-to-leave" information is only a rough estimation of the time that a parking spot can be released. In reality, sellers might leave earlier or later depending on various delays or just a poor estimation of their leaving time. Buyer arrival time can be similarly uncertain — traffic jam, detours, and other factors can make precise estimation of arrival time very hard. Uncertainty in sellers' departure and buyers' arrival times can cause two problems: 1) if buyers arrive later than the departure of sellers, there is a time window during which the reserved spot may be taken by *hidden drivers* i.e. drivers other than the buyer, and 2) if buyers arrive earlier than the departure of sellers, they have to wait, which is difficult in many locations since it can cause traffic backups. In the rest of this section, we discuss s these two problems respectively.

5.2.1 Buyer-Seller Gap

If a buyer arrives after than the seller departs, the buyer has to compete with hidden drivers. In this case, buyers have a lower chance of successful parking at the spot. There are two cases when a buyer can arrive after the seller leaves: 1) the seller leaves punctually or later than then reservation time, but the buyer arrives even later, and 2) the buyer arrives punctually or earlier than the reservation time, but the seller leaves even earlier.

Late Arrival Notice: In the first case, it is buyer's fault for not being able to meet the reservation time. However, it is very likely that the buyer is already on road and may be only a few blocks away from the parking location. In this case, if sellers are willing to wait for a few more minutes, the reservation can still happen successfully. CrowdPark uses a "*Late Arrival Notice*" service to co-ordinate between the buyer and seller.

CrowdPark calculates the estimated arrival time for buyers by utilizing their latest GPS location and the destination location, and determines whether this time will be later than the WTL time. In our evaluation in §7, we demonstrate that state-of-art navigators provide an estimated travel time with only a few minutes error within a radius of five miles from the destination. Therefore, the estimated travel time from navigators can be utilized as a indicator for sellers to decide how much longer to wait. The "Late Arrival Notice" service is designed as follows: if buyers enable the navigator-integrated in CrowdPark client for routing, for example the Nokia Ovi Map, CrowdPark estimates the arrival time directly on client side. Otherwise, the CrowdPark client sends the latest GPS locations back to server to estimate the travel time. The estimated travel time is sent to sellers in two cases, either when the seller decides to leave by sending a "leaving now" message to server, or when the estimated travel time exceeds the reservation time. In both cases, CrowdPark asks sellers if they are willing to wait until the estimated arrival time of buyers.

Seller Not In Scene: CrowdPark tries to discourage sellers from leaving earlier than the reservation time, since a buyer would have too little time to adjust to such an eventuality. If a seller leaves earlier than the reservation time and it results in a failed reservation for the buyer, we grant full refund to the buyer if they show evidence that they parked at a different location. Using the ActCheck mechanism running on the seller's phone, we can identify if the seller left prior to the WTL time. In addition, if the buyer provides evidence that they parked at a different location using a geo-tagged image of their license plate, we use the SpotCheck approach discussed in §4 to validate this proof and provide a full refund to the buyer. (If the seller persistently leaves prior to the expected time, the participation incentive paid to the seller may be potentially withdrawn as well to discourage such behavior.)

5.2.2 Buyer-Seller Overlap

If buyers arrive earlier than the departure time of sellers, buyers have to wait for sellers to release the parking spot. In some instances, buyers might be able to find a temporary spot near the reserved parking spot to park and wait for a short duration. We refer to this as "double parking". However, double parking time can be very short, typically one to two minutes in downtown street parking to avoid blocking traffic, or a few minutes within parking garages. After the double parking time, buyers have to circle around the reserved parking spot and return to check again if sellers are leaving or not. If sellers leave during the time buyers are circling, buyers again have to compete with hidden drivers, which reduces their chances of obtaining the slot.

To avoid buyers from losing the spot while circling, CrowdPark allows buyers to send an "arriving now" message via CrowdPark to ask seller for a updated "when-to-leave" message. If buyers decide to accept the new WTL, CrowdPark can give a guide of where to circle and estimate the circling time using GPS-based navigation techniques (similar to the "Late Arrival Notice" service described earlier). The estimated arrival time is sent back to seller. If sellers agree to the new arrival time, the reservation is still valid.

6. SYSTEM IMPLEMENTATION

6.1 CrowdPark System

The CrowdPark system consists of a backend server at Nokia Research at Palo Alto, and an HTML-5 based client designed for several popular smartphones. The components diagram of CrowdSearch system is shown in Figure 4.

CrowdPark Client: The client consists of two views, seller's view and buyer's view. The seller's view contains two major modules: reservation module and ActCheck module. The reservation module allows users to submit WTL messages to the CrowdPark server and get their reward, and ActCheck Module initiates malicious user detection after users send WTL messages or LN messages. The buyer's view also contains two major modules: reservation and navigation. The reservation module allows users to search and buy WTL messages, and the navigation module allows users to navigate to reserved parking spot and calculates the estimated travel time.

CrowdPark Server Implementation: The CrowdPark server comprises the incentive engine, reservation engine, and cooperation engine. The incentive engine calculates system parameters based on the constraints we derived in §3. The reservation engine accepts and validates WTL messages from sellers, and reservation requests from buyers. It is also responsible for rewarding sellers and refunding buyers, according to the system parameters determined by the incentive engine. The cooperation engine deals with the negotiation of a new WTL time when buyers send a Late-arrival Notice. Besides the three major engines, there are two other modules in CrowdPark server. One is the ANPR module, which runs an ANPR software for plate number image recognition. The other is the map service, which supports the navigation and map functions on the client.

Implementation Status: CrowdPark is still under prototyping, and we are working on a full live version of the system. Our current implementation of CrowdPark server contains over 10,000 lines of Java code. We have implemented the three engines and integrated with ANPR. Our next step is to integrate with Nokia's Ovi Map to provide map service and enable the navigation function on the client side. Our current implementation of the CrowdPark client contains over 5,000 lines of javascript code for the logic part and over 10,000 HTML-5 and CSS code for UI design. The reservation module is implemented on the client, but the ActCheck module is yet to be integrated since a cross-platform implementation is not currently available. We plan to have a cross-platform implementation of ActCheck (currently it is written in Symbian C++), and integrate it with other parts of the client.

6.2 CrowdPark Simulator

Besides the implementation of CrowdPark system, we also built a CrowdPark simulator to simulate the reservation performance for different parameter settings. The simulator consists of three major components: user engine, hidden user engine, and Reservation engine. The user engine generates sellers and buyers following a Poisson distribution. It also simulates parking events as well as transactions (i.e., buy, sell, and confirm) of WTL messages. The hidden user engine generates hidden drivers according to a Poisson distribution and their

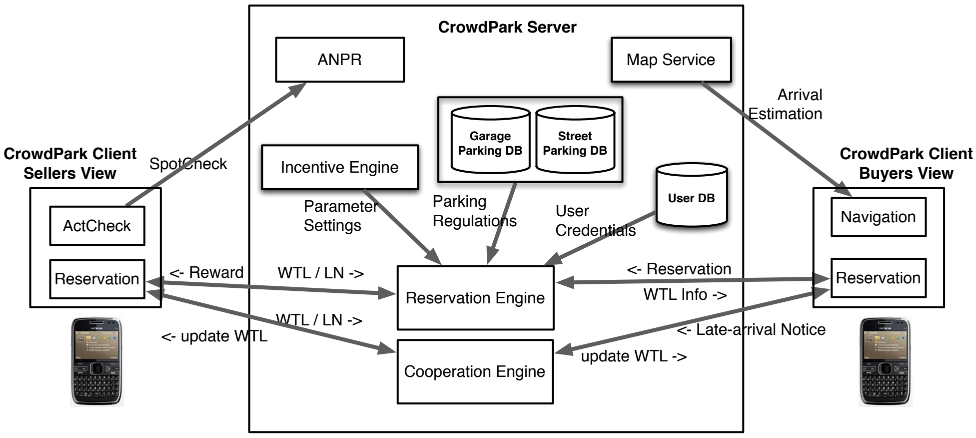


Figure 4: CrowdPark Architecture.

parking events. The reservation engine manages WTL messages for sale and rewards users according to the reservation results. The WTL lifetime is modeled in six states: CREATED, SOLD, EXPIRED, PARKED, CONFIRMED, and LOST, where LOST means that the spot is taken by hidden drivers.

The CrowdPark simulator is driven by events that occur sequentially. At every simulation step, the simulator first adds new buyers or sellers into queues managed by the simulator, then enumerates all buyers, sellers, and WTL messages. While visiting each object in the queues, the reservation engine updates the state of the corresponding WTL and the gains (or earnings) of each buyer and seller. We assume that user behavior is decided by a simple honesty attribute – either always honest or dishonest. This is clearly an approximation of human behavior but is sufficient for understanding CrowdPark performance.

7. EVALUATION

We evaluate incentive design, malicious seller detection, and handling uncertainty in this section.

7.1 Variation in Parking Availability

In this section, we evaluate the lifetime of empty parking spots in San Francisco downtown areas, where street-parking supply is far less than parking demand. The purpose of this study is two-fold: first, we use this study to demonstrate the parking problem in big cities, and second, the study provides valuable parameters to seed our simulator models.

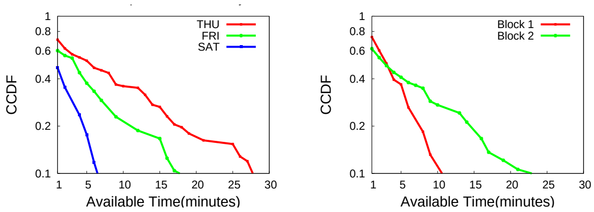
We spent two weeks in August 2010 monitoring street parking in San Francisco. We monitored four blocks in total — two blocks near Union Square and two blocks in the Financial District. We recorded the time at which each parking spot is taken and released for all street parking spots within these four blocks. Each day, we monitored the street parking from noon to 6 pm to cover both lunch time and rush hours.

The availability of parking exhibits significant temporal dependency. Figure 5(a) shows the log-scale CCDF of the lifetime of empty parking spots for three consecutive afternoons near Union Square, San Francisco. From this figure, we find that 80% of the time, an empty parking spot is taken within 5 minutes on Saturday afternoon. However 80% of empty parking spots has 10 minutes or longer lifetime on Friday afternoon, and around 15 minutes on Thursday afternoon. The numbers matches our observation: on Thursday afternoon, a large fraction of vehicles parked along streets are commercial ones, while both personal vehicles and commercial ones compete with each other to park along streets on Saturday.

Parking availability also varies spatially. Figure 5(b) shows the log-scale CCDF of the lifetime of empty parking spots in two adjacent blocks in Financial District during the same time period. Since one block is closer to shopping areas, it has significantly higher parking demand than the other. From this figure, we observe that the lifetime of empty parking spots in the busy block is about half that of the other block.

These results have two implications. First, the results show that systems that use solely parking availability information are limited due to their inability to precisely know the time when a spot is taken. For example, Google OpenSpot keeps a spot on the map for 20 minutes by default, but that number clearly depends on the location and time as described above. Second, the results help seed our simulator with appropriate models of hidden driver behavior for a parking spot. The plots are all roughly linear in logscale, which gives credence to the use of a poisson distribution for modeling hidden drivers. Also, we see that in heavily congested areas, the inter-arrival time of parking events is about 2-5 minutes, which we also use in our simulations.

7.2 Setting Reservation Parameters



(a) Temporal Behavior (b) Spatial Behavior

Figure 5: Spatial and Temporal Dependency of Parking Availability in San Francisco

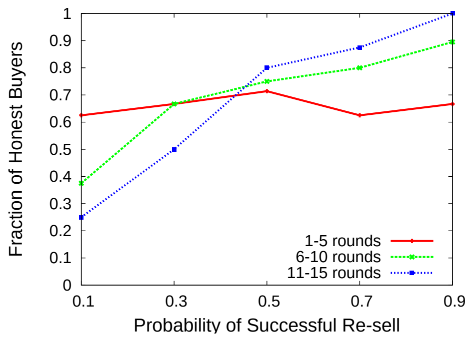


Figure 6: User Study of honest confirmation using Amazon Mechanical Turk.

In §3, we derived value ranges for base reward, bonus, and refund to ensure buyer honesty and profitability to the service provider. In this section, we study how to configure these parameters in real scenarios. Our evaluation consists of two parts. First, we evaluate the fraction of rational users via a user study and demonstrate the relation between reward and truth-telling. Next, we evaluate how a service provider can set the bonus despite the presence of dishonest users.

7.2.1 *User Study: Rational Behavior of Users*

The goal of this user study is to understand whether the parameter choices made by CrowdPark would lead to honest reporting from buyers. One of the assumptions that we make is that a buyer can quickly determine that honesty leads to the best outcome, and is therefore the most rational approach. But how quickly can a buyer identify that this is indeed the case?

We utilize Amazon Mechanical Turk (AMT) [24] in this evaluation. We recruit 131 participants from AMT for this study. We design a web-based survey that replicates the actual scenario of buyer confirmation as follows: participants assume that they have already paid \$2 for reserving a parking spot. Depending on whether they successfully park or not, they can either claim a partial refund of \$1 or re-sell the reserved parking spot. Re-selling gives them a base reward of \$0.2 and a bonus of \$2 if re-selling leads to another successful parking. For each participant, the probability that they can re-sell their parking spot successfully, denoted as p, is set a priori but hidden from participants. The re-selling probability is randomly chosen from a set of 0.1, 0.3, 0.5, 0.7, and 0.9 uniformly. Note that changing the probability p effectively changes the expected bonus pX. Also note that when the probability is 0.4, the gain of re-selling becomes same as partial refund. Rational users would choose to re-sell against refund for the probability greater than 0.4. In order to observe how participants converges to a final decision, we ask each participant to answer this survey question for 15 times. Each time the re-selling result is randomly generated based on the preset probability p. Since one of our goals is to test how quickly users can converge to the best strategy, we ask participants to maximize their gain during 15 rounds and give them a \$5 bonus for their achievements. We divide the 15 round responses into three groups: 1-5 rounds, 6-10 rounds, and 11-15 rounds in the following evaluation.

Figure 6 illustrates how users behave in terms of rationality when p varies. From this figure, we first find that higher reward significantly increases the fraction of users who choose tell the truth. We also see that people can learn the 0.4 threshold quickly. We observe that the fraction of honest confirmation remains almost constant for the first five rounds regardless of the amount of reward because users are trying to infer the underlying hidden probabilities. But after 15 rounds, people learn the best strategy: when p is 0.1, around 80% participants choose lying, which is the best strategy for a rational user in this case. However, when p approaches to 0.9, almost all participants choose honest confirmation, which is again the best strategy for a rational user for the case.

7.2.2 *Simulation Study: Setting Parameters*

We now study how to set bonus when the fraction of rational users (or honest confirmation) varies. In our simulation, we assume that the budget from service provider is \$0.1 per WTL message, and the price for a WTL message is \$2. Following the steps presented in the example in §3, we choose $D = \$0.2, R = \$0.3$, and $X = \$0.3 + \frac{\$0.2}{\hat{p}}$, where the value of $\hat{p}$ denotes the estimated value of p, and X is chosen to make the bonus meet the truth-telling constraint and budget constraint. In our simulator, the value of $\hat{p}$ is predicted by a service provider by counting the number of successful parking reservations over the previous 10,000 WTLs.

Figure 7 shows the lower and upper bounds for bonus and the actual bonus setting in our simulation. We vary the fraction of rational users from 1% to 100% with a step 1%, and for each fraction setting, we simulate 100,000 WTL transactions. This figure shows that

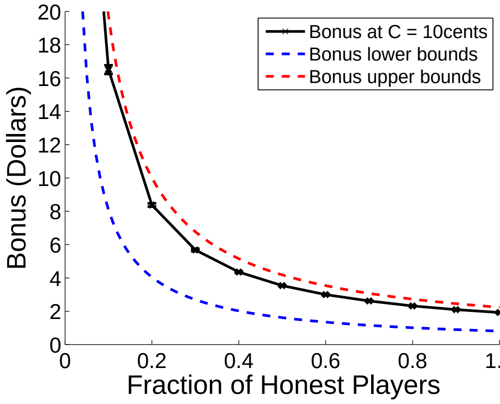


Figure 7: The bonus setting is dynamically changed as the fraction of rational buyers changes.

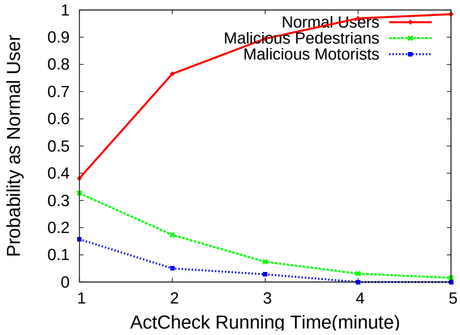


Figure 8: ActCheck accuracy vs. running time.

bonus is correctly set within its bounds despite the presence of dishonest users. Meanwhile, we can also conclude from this figure that when the fraction of honest buyers is over 50%, the value of bonus is relatively stable, implying that a fixed bonus is robust to variation in the fraction of honest buyers.

7.3 Detecting Malicious Sellers

7.3.1 Performance of ActCheck

In this section, we evaluate how accurately ActCheck can detect malicious sellers. We focus on two types of malicious sellers, pedestrians and motorists. We first evaluate the case where malicious sellers are unaware of ActCheck, and then evaluate the case where malicious sellers are aware of ActCheck and try to deceive it.

Malicious Users Unaware of ActCheck: We run ActCheck after sellers send either "when-to-leave" message or "leaving-now" message. For each case, we tune the running time of ActCheck from one minute to five minutes. Figure 8 shows the accuracy of ActCheck in classifying normal users and malicious users. From this figure, we find that the accuracy of ActCheck improves significantly as the running time increases. When ActCheck runs for five minutes, it can classify over 98% normal users correctly, while less than 5% malicious users can pass this classification filter. While the energy consumption of ActCheck increases with time, it is still energy-efficient in comparison to transferring an image since it only uses an accelerometer sensor [19].

Malicious Users Aware of ActCheck: In this experiment, we act as malicious sellers to deceive ActCheck. Since ActCheck expects that a seller behave as a pedestrian, a runner, or a cyclist after sending "when-to-leave" message and behaves as a motorist after sending the "leaving now" message, we tried two malicious behaviors in this experiment: 1) act as a pedestrian in a moving vehicle, and 2) act as a motorist when walking.

To act as a pedestrian in a moving vehicle, users need to meet two conditions: acceleration (in x and y axes) and high frequency vibration (caused by moving vehicles) should be kept small enough. To keep the acceleration value small enough, users have to drive smoothly and slow. To force the high frequency vibration, users have to detach the phone from vibrating objects (i.e., vehicle and drivers); users can throw and catch the phone in vehicles. ActCheck can be deceived if both conditions are met. However, the effort to perform such deception becomes harder particularly when the running time of ActCheck is sufficiently long. To act as a motorist when walking is much harder, since it is difficult to mimic the high frequency vibration pattern of vehicles.

7.3.2 Performance of SpotCheck

In this section, we evaluate the performance of SpotCheck. We collected 54 license plate images using mobile phones. Collected images cover a large spectrum of real-world conditions such as different lighting, exposure, and focus. We use an open-source Automatic Number Plate Recognition (ANPR) software called javaANPR [20] to recognize the plate numbers automatically. We also send plate images to Amazon Mechanical Turk for human-based recognition. We recruit five different AMT users to recognize each plate image and pay each individual five cents to recognize a plate. We then evaluate the accuracy and delay of the first response, the majority of the first three responses, and the majority of all five responses.

Table 3 shows the performance of both ANPR and human validation. In this table, Human-k denotes that we evaluate the first k AMT responses. To our surprise, we found that ANPR software has very low accuracy.[1] However, AMT-based human recognition performs well

[1]We expect that commercial ANPR software such as those used by the Department of Transportation may have higher accuracy. However, these were too expensive for experiment.

# Methods	Recognition Accuracy	Average Delay(seconds)
ANPR	10%	5
Human-1	96%	98
Human-3	98%	220
Human-5	100%	325

Table 3: The accuracy and delay of using ANPR and Amazon Mechanical Turk for Vehicle Number Plate Recognition.

# Segments	Precision
1	0.64
2	0.85
3	0.89
4	0.95

Table 4: Accuracy of street-segment matching.

enough to be an excellent alternative. From Table 3, we find that the accuracy of human recognition is close to 100% even with a single individual. The delay is only one or a few minutes, which is sufficiently small for its use in CrowdPark. Thus, even without a professional ANPR software, AMT-based human recognition achieves high accuracy and acceptable delay for SpotCheck.

7.4 Handling Uncertainty

We now evaluate approaches that we use for handling uncertainty in location, seller departure time, and buyer arrival time.

7.4.1 Performance of Street-Segment Matching

In this section, we evaluate the performance of our street-segment matching algorithm. We conducted 30 trips in San Francisco downtown area, and 12 trips in San Jose downtown area to evaluate our algorithm. San Francisco downtown area is full of highrise buildings with cloudy weather, which gives the worst condition for GPS positioning performance. For each trip, we stop at a parking location for around two minutes to collect GPS samples. We sample GPS readings every five seconds, so we collect up to 24 GPS samples per parking location. For each trip, we run our street segment matching algorithm to find the most likely street segment. Table 7.4.1 shows the precision of our street segment matching algorithm over top-4 results. Here the precision of top-n results is defined as the probability that one of top-n results contains a correct match.

From Table 7.4.1, we find that street-segment matching is a reasonable heuristic for capturing the location of the parking spot under noisy GPS samples. If we augment this technique by asking the seller to choose the correct segment from four choices, then parking spot localization can be precise enough. In conjunction with other information such as vehicle physical information (e.g., car make/models/color) and the meter identifier discussed in §5.1, parking spot localization can be made precise as well.

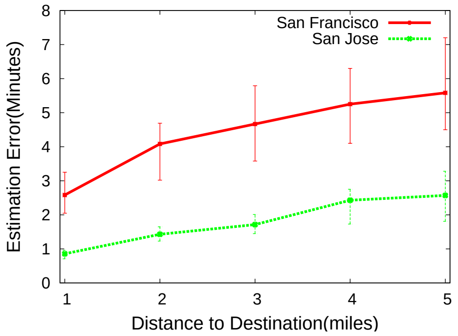


Figure 9: Accuracy of Late Arrival Notice. In this figure, positive value in y-axis means the estimated time is sooner than the actual travel time.

7.4.2 Performance of Late Arrival Notice

In this section, we evaluate the performance of the Late Arrival Notice that buyers send sellers. The goal of this experiment is to evaluate how accurately navigators can estimate the arrival time and trigger the Late Arrival Notice. We used two commercial navigators, Google map and Garmin, and average the results. We made 12 trips in the San Francisco downtown area, and 8 trips in San Jose downtown area. Each trip has a length of over five miles. San Francisco trips encounter more traffic lights and pedestrians than San Jose trips, which makes travel time more unpredictable.

Figure 9 shows the average error of estimated travel time (actual travel time minus estimated travel time) vs. distance. Clearly, the estimation error is higher when the distance is longer. From this figure, we can conclude that the estimation error from navigators is within a reasonable range for the Late Arrival Notice to be useful. First, we see that the absolute error when the buyer is about five miles away is around five minutes in San Francisco, and around two minutes in San Jose. It means that if a navigator estimates that a buyer can arrive on time, the actual case would be no worse than having a five minutes delay, which can be informed to the seller. In addition, this error continually reduces and becomes close to a couple of minutes when the buyer is about a mile away, which is typically the time at which a seller may be near their vehicle. Second, the estimated travel time is always smaller than the actual travel time. In other words, the navigators always under-estimate arrival time. The benefit is that there is no false alarms. When a Late Arrival Notice is triggered, it is always true.

7.4.3 Simulation Study: Performance of CrowdPark

The time gap between a seller's departure and a buyer's arrival can be either zero or greater. Zero gap can be achieved by punctual buyers and sellers on WTL time. Also, the Late Arrival Notice and "leaving now" messages can help reduce the time gap close to zero. In this section, we explore how the time gap impacts successful reservation rate. There are two cases that we need to study: 1) a buyer arrives after a seller leaves and 2) a buyer arrives before a seller leaves. In the former case, reservations can fail because of hidden drivers – a nearby driver can occupy the parking spot during the gap between a seller and a buyer. In the latter case, reservations can fail since a buyer can only wait at the parking location for limited time (e.g., double parking). We model the behavior of buyers under the situation as follows: if a buyer arrives earlier than the departure of seller, the buyer parks close to the reserved spot for a few minutes and circles if the spot is still not released. We call it double parking. The buyer comes back to the reserved spot after a few minute long circling and repeats the action of double parking if the seller has still not left. During the circling time of the buyer, the seller can leave and hidden drivers may occupy the spot.

We set the interarrival time of hidden drivers to be two to five minutes, as observed in our parking availability pattern study in §7.1. The circling time may change depending on situations (e.g., garage or on-street, traffic conditions, signalized intersections, and pedestrians). We model the circling time a uniform distribution, the interval of which is from three minutes to seven minutes. We consider double parking times of 30 seconds, 1 minute, and 2 minutes. The larger double parking time corresponds to a buyer at a parking garage.

Figure 10(a) illustrates how the successful reservation rate decreases as the gap between the seller and buyer increases when the seller leaves before the buyer. For example, if a buyer arrives within two minutes after a seller's departure and the rate of hidden drivers is 0.5 per minute, then the buyer's probability of successful reservation is 50%. Therefore, we need to encourage sellers to wait a few more minutes for buyers to reduce the chance of sellers' early departure, especially when WTL time has not been reached.

Figure 10(b) depicts the case when buyers arrive before sellers. The curves are periodic due to buyer circling. If sellers leave during the double parking time, buyers can successfully park. However, a hidden driver may park if the seller leaves during the circling time.

Cooperation between sellers and buyers significantly prevents periodic drops in reservation rate that are observed in figure 10(b). Figure 10(c) shows the benefit of cooperation in terms of successful reservation rate. If buyers simply notify sellers of their circling, sellers can wait for a few more minute depending on their tolerance levels. With only one minute long waiting, the reservation rate can be increased by 25% in average in comparison with no cooperation. If sellers can wait for five more minutes, the reservation rate is over 90% and two times better than when there is no cooperation.

8. RELATED WORK

We now provide an overview of salient related work.

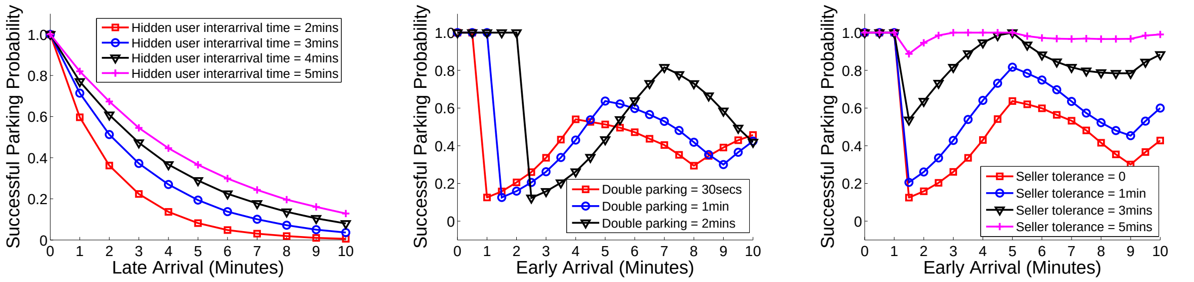
Crowdsourcing Challenges: There has been a growing body of work that address various challenges in crowdsourcing including the micro-payment models [13, 25, 33], auction-based models [7, 17], and data quality issues [29, 32]. Our work is unique in that it looks at all of these issues in conjunction — we combine incentive design and sensing techniques together to ensure users providing high quality data.

Crowdsourced Parking Availability: An increasing number of mobile crowdsourcing applications allow users to share empty parking spot information. Examples include OpenSpot [14], Rodify [15], and others [3, 16, 26]. Among all these applications, Rodify is closest to us, as it allow users to share parking spots that are both available now and available soon. However, none of these applications solve the challenges we addressed in this paper, including incentives, detecting malicious users, and handling real-world uncertainties. Another relevant recent project is ParkNet [21], which installs ultra-sonic sensors on vehicles, and detects parking availability when vehicles drive by. This approach requires expensive infrastructure to be installed, and suffers from the same limitation as other approaches in that it only provides availability information and not when a spot is taken, unlike our approach.

Participatory Sensing Applications: There have been a spectrum of participatory sensing and personal sensing applications for mobile phones that use onboard sensors to obtain bike routes [8], images [4], activity patterns [18], bus arrivals [30], traffic and road quality [9, 10, 12], and others [11, 22]. CrowdPark leverages activity monitoring mechanisms that many of these approaches propose. We differ from these approaches by integrating participatory sensing with incentive mechanisms and data authenticity checking techniques.

9. CONCLUSION

Crowdsourcing has seen considerable interest in recent years, and has been seen as a potential solution for a wide range of technical problems including parking in urban areas. However, realizing the promise of crowdsourcing necessitates that we tackle thorny technical problems including incentive design, authenticity of information, precise localization, and a plethora of other real-world issues. This paper provides a holistic solution that addresses several of these issues in the



(a) Late Arrival (b) Early Arrival (c) Early Arrival with Cooperation

Figure 10: Reservation successful rate when buyers arrive later and earlier than seller departure.

context of a parking reservation system using mobile phones. We address the challenges using a combination of incentive protocol design, game-theoretical and cost-benefit analysis, and sensing based techniques. While this paper presents a specific combination of approaches to address the parking problem, we believe that the methodology that we follow is more widely applicable to other mobile crowdsourcing applications.

We address several practical challenges but many more remain. We need to further study the spatial and temporal dynamics of parking, have better characterization of human behavior in a parking reservation system, have better integration with existing parking information systems, and so on. However, our efforts demonstrate that crowdsourced parking reservations is a viable model for addressing a pressing societal need.

10. REFERENCES

[1] Facebook credits. `http://www.facebook.com/credits/`.

[2] R. Arnott and E. Inci. An integrated model of downtown parking and traffic congestion. *Journal of Urban Economics*, 60:418–442, November 2006.

[3] N. Bilton. Finding That Prime Parking Spot With Primospot. `http://bits.blogs.nytimes.com/2009/12/03/finding-that-prime-parking-spot-with-primospot/`.

[4] N. Bulusu, C. Chou, and S. Kanhere. Participatory Sensing in Commerce: Using Mobile Camera Phones to Track Market Price Dispersion. In *UrbanSense*, 2008.

[5] D.-K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla. Autogait: A mobile platform that accurately estimates the distance walked. In *Percom 2010*, pages 116–124.

[6] J. V. Derbeken. Fatal stabbing over parking. `http://articles.sfgate.com/2006-09-19/bay-area/17312921_1_parking-space-parking-spot-stabbed`.

[7] D. DiPalantino and M. Vojnovic. Crowdsourcing and all-pay auctions. In *ACM EC*, pages 119–128. ACM, 2009.

[8] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. S. Ahn, and A. T. Campbell. BikeNet: A Mobile Sensing System for Cyclist Experience Mapping. 6, 2009.

[9] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *ACM MobiSys*, 2008.

[10] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. In *ACM MobiSys*, pages 151–164, 2010.

[11] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *ACM MobiSys*, 2008.

[12] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *ACM Mobisys*, 2008.

[13] B. A. Huberman, D. M. Romero, and F. Wu. Crowdsourcing, attention and productivity. *CoRR*, 2008.

[14] J. Kincaid. Googles Open Spot Makes Parking A Breeze, Assuming Everyone Turns Into A Good Samaritan. `http://techcrunch.com/2010/07/09/google-parking-open-spot/`.

[15] N. Lamba. Social Media Tackles Traffic. `http://www.wired.com/autopia/2010/12/ibm-thoughts-on-a-smarter-planet-8/`.

[16] J. Lee. An App Gives a Heads-Up on Parking Spaces. `http://cityroom.blogs.nytimes.com/2009/12/03/an-app-gives-a-heads-up-on-parking-spaces/`.

[17] J. Lee and B. Hoh. Sell your experiences: A market mechanism based incentive for participatory sensing. In *Percom*, 2010.

[18] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *ACM Mobisys*, 2009.

[19] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *ACM Sensys*, November 3-5 2010.

[20] O. Martinsky. Java anpr. `http://javaanpr.sourceforge.net/`.

[21] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. Parknet: drive-by sensing of road-side parking statistics. In *ACM MobiSys*, 2010.

[22] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *ACM SenSys*, 2008.

[23] U. D. of Transportation. Advanced parking management systems: A cross-cutting study. `www.its.dot.gov/jpodocs/repts_te/14318_files/14318.pdf`.

[24] G. Paolacci, J. Chandler, and P. Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5, August 2010.

[25] S. Reddy, D. Estrin, M. Hansen, and M. Srivastava. Examining micro-payments for participatory sensing data collections. In *ACM Ubicomp*, pages 33–36, 2010.

[26] W. Roush. Can SpotScout Take the Pain out of Parking? `http://www.xconomy.com/boston/2008/02/12/can-spotscout-take-the-pain-out-of-parking/`.

[27] SFMTA. SFPark - About the Project. `http://sfpark.org/about-the-project/`.

[28] D. Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, November 2006.

[29] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263, 2008.

[30] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *ACM Sensys*, pages 85–98, 2010.

[31] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *ACM MobiSys*, pages 179–192, 2009.

[32] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *ACM Mobisys*, pages 77–90, 2010.

[33] J. Yang, L. A. Adamic, and M. S. Ackerman. Crowdsourcing and knowledge sharing: strategic user behavior on taskcn. In *ACM EC*, pages 246–255, 2008.