Participatory Sensing Meets Opportunistic Sharing: Automatic Phone-to-Phone Communication in Vehicles

Xiaoshan Sun, Shaohan Hu, Lu Su, *Member, IEEE*, Tarek F. Abdelzaher, Pan Hui, Wei Zheng, Hengchang Liu, and John A. Stankovic, *Fellow, IEEE*

Abstract—This paper explores direct phone-to-phone communication (via WiFi interface) among vehicles to support participatory sensing applications. Sensing data usually contains location, speed, and fuel consumption of the car, and has a long time delay between collected and transferred to the server. Direct communication among phones aboard is important in reducing data transfer delay time and sharing participatory sensing information in an inexpensive manner. We design a practical and optimized communication mechanism for direct phone-to-phone data transfer among phones aboard that strategically enables phone-to-phone and/or phone-to-WiFiAP communications by optimally toggling the phones between the normal client and the hotspot modes. We take advantage of the WiFi hotspot functionality on smartphones, and hence require neither involvement of participants nor changes to existing wireless infrastructure and protocols. An analytical model is established to optimize toggling between client and hotspot modes for optimal system efficiency. We fully implement this system on off-the-shelf Google Galaxy Nexus and Nexus S phones. Through a 35-vehicle two-month deployment study, as well as simulation experiments using the real-world T-drive 9,211-taxicab dataset, we show that our solution significantly reduces data transfer delay time and maintains over 80 percent system efficiency under varying system parameters.

Index Terms—Phone-to-phone communication, vehicular networking, WiFi hotspot, parameter optimization

1 Introduction

This paper presents a practical mobile phone sensing system that utilizes direct phone-to-phone communication between vehicles to improve performance of mobile participatory sensing applications. Rather than designing a new protocol to improve vehicle-to-vehicle and vehicle-to-WiFiAP communications (e.g., see work on delay/disruption tolerant networks (DTN) [1], [2], [3], [4], mobile ad-hoc networks (MANET) [5], [6], and vehicular networks [7], [8], [9]), we present an optimized phone-to-phone communication scheme that uses only those capabilities exported to the user on today's smartphones. It strategically toggles between the normal (client) and hotspot modes on smartphones as would be needed to collect data from phones and upload to a remote back-end server. It does so without needing to root or jailbreak smartphones, which makes the functionality implementable as a third-party phone application. Moreover, it requires neither involvement of participants nor changes to existing wireless infrastructure and protocols.

This work is motivated by the proliferation of sensor-equipped smartphones in the past few years. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, it is estimated that 982 million smartphones will be shipped worldwide in 2015 [10]. The rich set of embedded sensors on smartphones makes mobile phone sensing an useful paradigm to support many applications that require real-time situation awareness, such as monitoring traffic congestion and commute delays. Vehicles are becoming popular as carriers of mobile sensing platforms for many reasons. First, their natural mobility increases coverage for many participatory and social sensing applications [11], [12]. Second, our daily commute itself has become a target of many research efforts, such as those that aim to save fuel consumption [13], find available parking positions [14], avoid traffic jams or routes in bad condition [15], [16], [17], or share general road-side events [18]. Research communities have recently investigated incentive mechanisms [19] to attract more smartphone users into mobile sensing, developed solutions to preserve participants' privacy [20], and addressed the sparse deployment problem [21] when mobile sensing systems do not have a sufficient number of participants.

Accordingly, we envision a new brand of sensing applications that use driver's phones to share mobile

- *X. Sun and H. Liu are with the Department of Computer Science, University of Science and Technology of China, Hefei, China. E-mail: sxs1166@mail.ustc.edu.cn, hcliu@ustc.edu.cn.*
- *S. Hu, L. Su, and T. F. Abdelzaher are with the Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL 61801. E-mail: {shu17, zaher}@illinois.edu, lusu@buffalo.edu.*
- *P. Hui is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, the Telekom Innovation Laboratories, Berlin, Germany, and the Aalto University, Helsinki, Finland. E-mail: panhui@cse.ust.hk.*
- *W. Zheng is with the Department of Statistics, University of Wisconsin Madison, Madison, WI 53706. E-mail: 18.weizheng@gmail.com.*
- *J. A. Stankovic is with the Department of Computer Science, University of Virginia, Charlottesville, VA 22904. E-mail: stankovic@cs.virginia.edu.*

Manuscript received 24 Mar. 2015; revised 29 Oct. 2015; accepted 11 Nov. 2015. Date of publication 25 Nov. 2015; date of current version 29 Aug. 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2503752

sensory data among vehicles as well as with infrastructure servers. We assume that users will exploit their cellular data bandwidth to *download results* from such servers, such as real-time traffic speed maps. However, they will typically not want the same mobile sensing applications to use their cellular communication for altruistic *raw data upload* to the server, since unlimited data plans are no longer prevalent [22], [23]. Instead, the paper explores a WiFi-based approach for uploading the sensor data needed for the service.

Wi-Fi based store and forward of sensed real-time data may result in a large latency [24], which motivates optimizing data transfer among vehicles as well as between vehicles and the infrastructure for faster offloading. Current communication techniques on smartphones that support peer-to-peer sharing, such as WiFi ad-hoc mode [25] and WiFi Direct [26], have significant limitations and are not directly usable for mobile sensing. WiFi ad-hoc is not supported on most popular phones unless rooted or jailbroken and will probably not be in the near future due to economic and political issues [27]. WiFi Direct was not designed with opportunistic networking in mind, but tries to connect WiFi enabled devices such as printers and cameras in a secure way and as easily as possible. User involvement is mandatory for WiFi Direct for security reasons [28]. Also note that even if WiFi Direct can overcome its mentioned limitations in the near future, the phones still need to switch between the WiFi Direct "peer mode" (to connect directly with other peers also in the peer mode) and the normal WiFi client mode (to connect to WiFi APs), as a phone in peer mode is not able to connect to normal WiFi APs to offload data. Thus our method actually generalizes to cover the WiFi Direct type of scenarios in the future.

In contrast, we utilize a WiFi hotspot switching approach that is compatible with existing WiFi APs as the functionalities needed are supported by the standard Android API and Java Reflection, which does not require users to root or jailbreak smartphones. Two phones can establish connections when one of them is in the *hotspot* mode and the other in the *client* mode, and a phone can offload data to access points when in the client mode. Initial efforts provided proof-of-concept prototypes [28]. Two important questions remain unanswered: first, is automatic phone-to-phone data transfer achievable in a highly mobile vehicular environment? Second, how to switch between the hotspot and client modes in an efficient way in order to minimize the expected wasted time due to phones being in incompatible modes? Our paper addresses the above questions, and makes the following contributions.

- In this paper, we present a fully deployed smartphone-based vehicular mobile sensing system in which automatic phone-to-phone communication is achieved and is compatible with existing wireless infrastructure. While social sensing regarding traffic and daily commutes provides the motivating applications, this paper is strictly about the mobile communication platform needed to support such applications.
- An analytical model is established to optimize system parameters in an adaptive fashion to achieve high system efficiency, which means the ratio of transfer time to meeting time of two cars (or a car and a WiFi AP) and will be well explaned in Section 3. We also provide empirical results to support several important design decisions in our system.
- We evaluate our analytical model and demonstrate the performance of our system by providing results from a real 35-participant two-month deployment using Google Android phones, as well as simulation experiments using T-drive 9,211-taxicab dataset [29], [30]. Results show that our solution significantly reduces data transfer delay time and maintains above 80 percent efficiency under varying system parameters, even achieving 90 percent for parameter settings of the latest smartphones.

The remainder of this paper is organized as follows. After discussing related work in Section 2, we give detailed problem descriptions in Section 3. We then present our analytical model and system designs in Sections 4 and 5. We evaluate our system and solution in Section 6. Finally Section 7 concludes.

2 Related Work

Prior work on vehicular mobile sensing and communication generally falls into one of two categories: either using phones for data collection and uploading (to back-end servers) without peer-to-peer communication; or using DTN- or MANET- style vehicle-to-vehicle communication but on dedicated hardware instead of phones. We are the first to offer a fully deployed system that leverages both phone-to-phone and phone-to-AP communications from vehicle-resident smartphones, customized for the needs of mobile sensing.

Several prior mobile social sensing applications leverage smartphones placed in vehicles. For example, the Nericell project [16] presents a system that performs rich sensing using smart phones that users carry with them in normal courses, to monitor road and traffic conditions. The GreenGPS system [13] provides a service that computes fuel-efficient routes for vehicles between arbitrary end-points, by exploiting vehicular sensor measurements available through the On Board Diagnostic (OBD-II) interface of the car and GPS sensors on smartphones. SignalGuru [15] is a software service that relies solely on a collection of mobile phones to detect and predict the traffic signal schedule, producing a Green Light Optimal Speed Advisory (GLOSA). These systems rely on WiFi access points, since transmitting data through cellular data networks is expensive. However, open public WiFi is becoming less prevalent as more access points are becoming private or secure. Our paper aims to overcome this drawback by allowing smart phones to exchange data in an opportunistic way to maximize upload opportunities.

Our application scenario requires moving wireless nodes and sometimes information processing in intermittently-connected networks. MANETs and DTNs are therefore important overlapping fields of research to our paper.

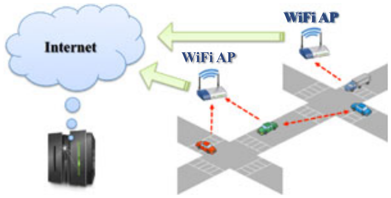


Fig. 1. System model.

For instance, CafNet [6] in the CarTel project [18] is a delay-tolerant stack that enables mobile data muling and allows data to be sent across an intermittently connected network. The CafNet protocols allow cars to serve as data mules, delivering data between nodes that are otherwise not connected to one another. Similarly, the DieselNet testbed [5] consists of 35 buses, each with a Diesel Brick, which is based on a HaCom Open Brick computer. Multi-Nets [31] investigates the switching between WiFi and cellular modes on phones for energy and/or throughput considerations. It is, however, not suitable for our targeted vehicular mobile sensing/networking scenarios because of limited WiFi accessibility in outdoor environments and that we do not allow cellular data transmission due to the constant generation of potentially huge amount of sensory data. Other related work in this field include [1], [2], [3], [4]. The main differences of our proposed system over this work are two-fold. First, most of them use data mules for data collections, instead, we systematically investigate the performance of realistic opportunistic networking via direct phone-to-phone communication, which is now possible with most popular mobile devices. Second, while they mainly focus on the optimization of communication stack to take advantage of short vehicle meeting times, we aim to leverage commonly open APIs on smartphones and hence restrict ourselves to what can be done with the available stacks.

Our work is also related to efforts in the vehicle networking community, called VANET, where the goal is usually to increase road safety and transport efficiency, and provide Internet access on the move to ensure wireless ubiquitous connectivity. Research challenges in evolving connected vehicle architecture, such as leveraging street parking to enable vehicular Internet access [7] and investigating application-driven inter-and intra-cluster communication in VANETs [8], has been deeply investigated. However, in mobile participatory sensing, the vehicle-to-vehicle communication problem targets a different goal: we aim to help participants who rarely approach wireless access points themselves to deliver their sensory data to the back-end server more quickly. There appears to be no straightforward solution in the VANET regime to provide automatic and efficient vehicle-to-vehicle communication with smartphones.

Finally, existing communication techniques on smartphones that support peer-to-peer sharing, such as WiFi ad-hoc [25] and WiFi Direct [26], have significant limitations and are not directly usable for social sensing. WiFi Ad-Hoc is still not supported on most popular phones unless rooted or jailbroken and will probably not be in the near future [27]. WiFi Direct is not designed with opportunistic networking in mind, but tries to connect WiFi enabled devices such as printers and cameras in a secure way and as easily as possible [28]. In addition, once a phone is set to WiFi ad-hoc or Direct mode to support peer-to-peer communication, it is no longer able to connect to WiFi APs and offload data to the back-end servers. Our WiFi hotspot switching approach overcomes these drawbacks and does not need to root or jailbreak smartphones [28], however, there is still a lack of real deployment for performance evaluation especially in highly mobile environment, which incidentally is one main contribution of our paper as well.

3 System Model and Problem Description

Our system is aimed to operate in a vehicular mobile sensing network where sensory data is generated and collected from participants' vehicle-resident smartphones, as illustrated in Fig. 1. the system is composed of n cars, each with a smartphone inside to perform peer to peer data exchange as well as data upload into the Cloud/web server. WiFi coverage is only sparsely available within the sensing area. When a car moves into the coverage area of a WiFi access point, the phone transmits its locally stored data to the back-end server via WiFi communications. Moreover, we particularly allow phones to communicate with each other in order to reduce data transfer delays. Note that privacy and security issues are beyond the scope of this paper.

While a significant car density may be observed in an urban area, it may not be appropriate to assume that all or even a large portion of drivers are running our system on their phones. Instead, we make the more conservative assumption that only a small fraction of phones are running our system at any given time. Hence, it would be unusual for more than two such phones to be within each other's communication range at a time. Therefore, in this work we focus our analysis on pairwise encounters between phones, as opposed to optimizing general multi-party communications within phone clusters. To demonstrate the validity of our assumption, we record the number of vehicles in all meeting events in the T-drive dataset containing 9,211 taxicabs. We set the transmission range to be 30 m, according to our own transmission tests using Google smartphones in vehicles. We find that pairwise encounters make up about 80 percent of all meeting occurrences. Considering that the scale of this dataset is already quite large, the ratio of pairwise encounters would further increase with less participants in realistic settings.

In our system, as a phone joins the vehicle network, it enters the client mode, in which it searches for available communication opportunities, with either a phone in hotspot mode or a WiFi AP. Meanwhile, a timer is started to control how long the phone can stay in client mode searching. When the timer expires, the phone switches itself to become a WiFi hotspot. The phone then listens for incoming connection attempts from other phones that are in client mode. Similarly, another timer is used to switch the phone back to client mode upon expiry.

In either client or hotspot mode, whenever the phone sees a communication opportunity, the timer pauses as the phone enters transmission mode, and the data exchange

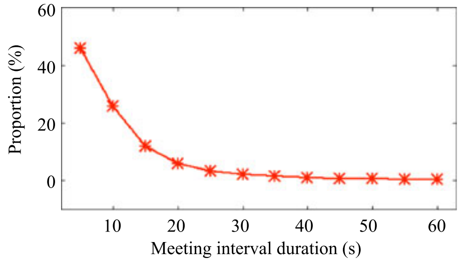


Fig. 2. Proportion of various meeting interval length from T-drive dataset.

starts with the other party (phone or back-end server via WiFi AP). When the communication is terminated due to either data transmission completion or cars moving out of range, the phone goes back to its previous mode, with the timer resumed.

As two phones approach each other, if they are both in hotspot (or client) mode, they cannot communicate until one of them toggles mode. Similarly, when a phone enters an WiFi AP coverage area, it cannot offload data if it's in hotspot mode. Therefore, the time durations phones stay in each mode is crucial. Under our described model, we are then interested in solving the *System Efficiency Optimization* problem, where *System Efficiency* is defined to be, of the entire time duration that phones are within communication range with each other (or WiFi APs), the proportion of time when data transfer can actually take place. The problem is challenging because the information when vehicles meet each other or move into WiFi coverage area is NOT *a priori*. In the next section, we establish an analytical model for the optimal mode-toggling policy and provide our solution to optimize important system parameters.

4 Analytical Formulation and Solution

In this section, we present the analytical formulation of the optimal mode-toggling policy for maximizing the total expected transmission duration in our targeted vehicular phone-to-phone networks.

We learn from preliminary experiments that connection rarely establishes in highway driving scenarios, regardless of whether the two cars are moving towards the same or opposite directions. On the other hand, when two cars meet and move toward the same direction in an urban or residential area, data transfer duration typically lasts quite long, which can also occur, for example, when the two cars close to each other park in the same parking lot or are caught in a traffic jam. Therefore, in these cases where the transmission duration is either extremely short or long, the switching of the phones' modes does not play a dominating role in system efficiency. Fig. 2 shows the distribution of car meeting interval lengths within the T-drive dataset. We observe that around 46 percent of meeting events last less than 5 seconds, and less than 1 percent longer than 1 minute. Thus, more than 50 percent of car meeting events are around the middle of the distribution and potentially can benefit considerably from our system.

A complete cycle of the hotspot switching procedure is decomposed in Fig. 3. As seen, a phone switches from client to hotspot mode with an overhead of t_0 seconds, stays in hotspot mode for r seconds, switches back to client mode with

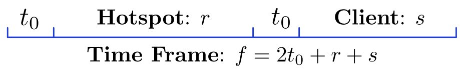


Fig. 3. Hotspot-Client switching cycle.

another t_0-second overhead, and then stays in client mode for s seconds, so on and so forth. Phones can retrieve the optimal mode-switching parameters from a central server.

For simplicity we assume that the hotspot-to-client and client-to-hotspot switching overheads are the same, confirmed by our experiments. Given the previous description, the switching procedure repeats with a period of $2t_0 + r + s$ seconds. Assuming the vehicle-vehicle and vehicle-AP meeting rates are β and γ, respectively ($\beta + \gamma = 1$), we have the following optimization objective function,

$$F(\beta, \gamma) = \max_{r,s}(\beta \cdot T_1 + \gamma \cdot T_2),$$

in which T_1 and T_2 are the expected phone-to-phone and phone-to-AP transmission durations, respectively. The optimization is over design parameters r and s. Other parameters are not design parameters.

Let the base function $f(t)$ be a periodic function with period $f = 2t_0 + r + s$,

$$f(t) = \begin{cases} 0, & 0 \le t \le t_0 \\ 1, & t_0 < t \le t_0 + r \\ 0, & t_0 + r < t \le 2t_0 + r \\ -1, & 2t_0 + r < t \le 2t_0 + r + s \end{cases}$$

When $f(t) = 0$, the phone is switching between modes; $f(t) = 1$ indicates that the phone is in hotspot mode, and $f(t) = -1$ client mode. In our calculation, we assume that the switching overhead does not dominate either of the actual mode durations, i.e., $r, s > t_0$.

Upon entering the communication range of each other, the two vehicles v_1 and v_2 are at t_1 and t_2 withint their respective base function $f(t)$ periods, $0 \le t_1, t_2 < 2t_0 + r + s$. We describe the switching patterns of v_1 and v_2 as $f_1(t) = f(t + t_1)$ and $f_2(t) = f(t + t_2), t \in [0, 2t_0 + r + s)$, respectively. We then define t^* to be the time since meeting that the phones in two cars establish connection. It follows that,

$$t^* = \min_t\{t : f_1(t)f_2(t) < 0\}.$$

Since two cars can meet at any time, we consider t_1 and t_2 to be uniformly distributed over $[0, 2t_0 + r + s)$. We use M_1 to denote the total time duration in which the two cars v_1 and v_2 are within communication range with each other. It then follows that the two phones can only establish connection if $M_1 \ge t^*$. With these notations, we derive the analytic formula of the expected transmission time

$$T_1 = E_{t_1,t_2}[M_1 - t^*],$$

which can be computed generically as

$$\frac{1}{f^2}\int_{t_1}\int_{t_2}(M_1 - t^*)dt_2 dt_1.$$

TABLE 1
Case Analysis for Expected Transmission Times

t_1	t_2	t^*	M_1	$E(M_1 - t^*)$
$[0, t_0)$	$[0, t_0)$	∞	$\sim$	0
$[0, t_0)$	$[t_0, t_0 + r)$		$M_1 \geq t_0 + r$	$\frac{1}{2f^2}\left[(M_1 - t_0)^2 - \frac{1}{3}(M_1 - r)^3 + \frac{1}{3}(M_1 - t_0 - r)^3\right]$
or		$2t_0 + r - t_2$	$r \leq M_1 < t_0 + r$	$\frac{1}{2f^2}\left[(M_1 - t_0)^2 t_0 - \frac{1}{3}(M_1 - r)^3\right]$
$[t_0, t_0 + r)$	$[0, t_0)$		$t_0 \leq M_1 < r$	$\frac{t_0}{2f^2}(M_1 - t_0)^2$
$[0, t_0)$	$[t_0 + r, 2t_0 + r)$	$\max(t_0 - t_1, 2t_0 + r - t_2)$	$M_1 \geq t_0$	$\frac{2}{f^2}\left(\frac{M_1}{2} t_0^2 - \frac{1}{3} t_0^3\right)$
			$M_1 < t_0$	$\frac{1}{3f^2} M_1^3$
$[0, t_0)$	$[2t_0 + r, f)$	$t_0 - t_1$	$M_1 \geq t_0$	$\frac{1}{f^2}\left[\frac{1}{3} t_0^3 - \frac{1}{2}(M_1 + s) t_0^2 + M_1 s t_0\right]$
			$M_1 < t_0$	$\frac{1}{f^2}\left(\frac{1}{2} s M_1^2 - \frac{1}{6} M_1^3\right)$
$[t_0, t_0 + r)$	$[t_0, t_0 + r)$	$2t_0 + r - \max(t_1, t_2)$	$M_1 \geq r$	$\frac{2}{f^2}\left[\frac{r - t_0}{2}(M_1 - t_0)^2 - \frac{1}{6}(M_1 - t_0)^3 + \frac{1}{6}(M_1 - r)^3\right]$
			$t_0 \leq M_1 < r$	$\frac{1}{f^2}\left[(r - t_0)(M_1 - t_0)^2 - \frac{1}{3}(M_1 - t_0)^3\right]$
$[t_0, t_0 + r)$	$[t_0 + r, 2t_0 + r)$	$2t_0 + r - t_2$	$M_1 \geq t_0$	$\frac{1}{f^2}\left[\frac{r}{2} M_1^2 - \frac{r - t_0}{2}(M_1 - t_0)^2 - \frac{1}{6} M_1^3 + \frac{1}{6}(M_1 - t_0)^3\right]$
			$M_1 < t_0$	$\frac{1}{f^2}\left(\frac{r}{2} M_1^2 - \frac{1}{6} M_1^3\right)$
$[t_0, t_0 + r)$	$[2t_0 + r, f)$	0	$\sim$	$\frac{M_1 r s}{f^2}$

Now we compute the actual analytical expression for T_1 for the various cases of t_1 and t_2 value ranges.

(I) We first consider the case where $t_1, t_2 \in [0, t_0)$. It's easily seen that $f_1(t) f_2(t) \geq 0$ for any t. Therefore, $E(M_1 - t^*) I_{(t_1 \in [0, t_0))} I_{(t_2 \in [0, t_0))} = 0$. Note that the I's are just indicator functions, which we omit in writing for the rest of the derivations.

(II) Next we consider the case where $t_1 \in [0, t_0)$, $t_2 \in [t_0, t_0 + r)$. Analyzing the physical process, we have $t^* = 2t_0 + r - t_2$ if and only if $t_2 - t_1 \geq t_0$. Then, if $M_1 \geq t_0 + r \geq t^*$, we have

$$
\begin{aligned}
&E(M_1 - t^*) \\
&= \frac{1}{f^2} \int_0^{t_0} \int_{t_1 + t_0}^{t_0 + r} (M_1 - 2t_0 - r + t_2) dt_2 dt_1 \\
&= \frac{1}{2f^2}\left[(M_1 - t_0)^2 t_0 - \frac{1}{3}(M_1 - r)^3 + \frac{1}{3}(M_1 - t_0 - r)^3\right].
\end{aligned}
$$

Alternatively if $r \leq M_1 < t_0 + r$, to ensure $M_1 \geq t^* = 2t_0 + r - t_2$, we need $t_2 \geq 2t_0 + r - M_1$. Now t_2 has two possible lower bounds, $t_1 + t_0$ and $2t_0 + r - M_1$. If $t_1 + t_0 > 2t_0 + r - M_1$, it can be inferred that $t_1 > t_0 + r - M_1$, then,

$$
\begin{aligned}
E(M_1 - t^*) &= \frac{1}{f^2} \int_{t_1} \int_{t_2} (M_1 - t^*) dt_2 dt_1 \\
&= \frac{1}{f^2} \int_{t_0 + r - M_1}^{t_0} \int_{t_1 + t_0}^{t_0 + r} (M_1 - 2t_0 - r + t_2) dt_2 dt_1 \\
&= \frac{1}{2f^2}\left[(M_1 - t_0)^2 (M_1 - r) - \frac{1}{3}(M_1 - r)^3\right].
\end{aligned}
$$

On the other hand if $t_1 + t_0 < 2t_0 + r - M_1$, it can be inferred that $t_1 < t_0 + r - M_1$, then,

$$
\begin{aligned}
E(M_1 - t^*) &= \frac{1}{f^2} \int_{t_1} \int_{t_2} (M_1 - t^*) dt_2 dt_1 \\
&= \frac{1}{f^2} \int_0^{t_0 + r - M_1} \int_{2t_0 + r - M_1}^{t_0 + r} (M_1 - 2t_0 - r + t_2) dt_2 dt_1 \\
&= \frac{1}{2f^2}(M_1 - t_0)^2 (t_0 + r - M_1).
\end{aligned}
$$

Adding these two together, we thus have,

$$
E(M_1 - t^*) = \frac{1}{2f^2}\left[(M_1 - t_0)^2 t_0 - \frac{1}{3}(M_1 - r)^3\right].
$$

If $t_0 \leq M_1 < r$, to ensure $M_1 \geq t^* = 2t_0 + r - t_2$, we have $t_2 \geq 2t_0 + r - M_1 \geq t_1 + t_0$, then

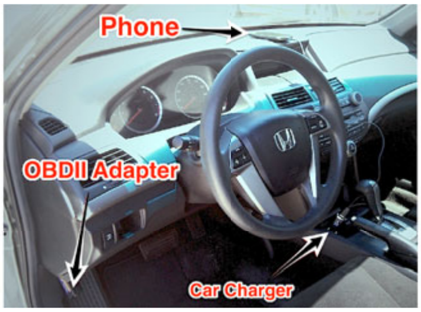
$$
\begin{aligned}
E(M_1 - t^*) &= \frac{1}{f^2} \int_{t_1} \int_{t_2} (M_1 - t^*) dt_2 dt_1 \\
&= \frac{1}{f^2} \int_0^{t_0} \int_{2t_0 + r - M_1}^{t_0 + r} (M_1 - 2t_0 - r + t_2) dt_2 dt_1 \\
&= \frac{t_0}{2f^2}(M_1 - t_0)^2.
\end{aligned}
$$

We omit derivation details for the rest of the cases and collect results for all cases in Table 1.

Since $[0, r + t_0)$ and $[r + t_0, f)$ are symmetric, we have

$$
\begin{aligned}
T_1 &= E[M_1 - t^*] \\
&= \frac{2rs}{f^2} M_1 + I_{(M_1 < t_0)} f_1(r, s, M_1) + I_{(M_1 \geq t_0)} f_2(r, s, M_1) \\
&\quad + I_{(t_0 \leq M_1 < r + t_0)} f_3(r, s, M_1) + I_{(M_1 \geq r + t_0)} f_4(r, s, M_1) \\
&\quad + I_{(t_0 \leq M_1 < s + t_0)} f_5(r, s, M_1) + I_{(M_1 \geq s + t_0)} f_6(r, s, M_1),
\end{aligned}
$$

(a) Hardware components of our prototype system



(b) Our system as installed in a vehicle

(c) System running screenshot

Fig. 4. Hardware components, installation, and running screenshot of our prototype system.

where

$$f_1(r,s,M_1) = \frac{s+r}{f^2}M_1^2$$

$$f_2(r,s,M_1) = \frac{1}{f^2}\left[\frac{2}{3}M_1^3 - \frac{2}{3}(M_1-t_0)^3 - 2t_0(M_1-t_0)^2 \right.$$
$$\left. + 2M_1t_0(r+s) + \frac{4}{3}t_0^3 - (2M_1+r+s)t_0^2\right]$$

$$f_3(r,s,M_1) = \frac{1}{f^2}\left[r(M_1-t_0)^2 - \frac{1}{3}(M_1-t_0)^3\right]$$

$$f_4(r,s,M_1) = \frac{1}{f^2}\left[r(M_1-t_0)^2 + \frac{1}{3}(M_1-t_0-r)^3 - \frac{1}{3}(M_1-t_0)^3\right]$$

$$f_5(r,s,M_1) = \frac{1}{f^2}\left[s(M_1-t_0)^2 - \frac{1}{3}(M_1-t_0)^3\right]$$

$$f_6(r,s,M_1) = \frac{1}{f^2}\left[s(M_1-t_0)^2 + \frac{1}{3}(M_1-t_0-s)^3 - \frac{1}{3}(M_1-t_0)^3\right].$$

Note that the conditions in the above equations are not mutually exclusive, which, however, does not affect the optimization.

For modeling phone-to-AP communication, assume that there is a WiFi AP always in hotspot mode. A vehicle comes in range of the AP and stays in range for M_2 seconds. With the same notations as above, we have

$$t^* = \begin{cases} t_0 - t_1, & t_1 \in [0, t_0) \\ 0, & t_1 \in [t_0, t_0 + r) \\ t_0 + s + (2t_0 + r - t_1), & t_1 \in [t_0 + r, 2t_0 + r) \\ t_0 + (f - t_1), & t_1 \in [2t_0 + r, f) \end{cases}.$$

Therefore, the expected connection time to a WiFi AP is,

$$\begin{aligned} T_2 &= E(M_2 - t^*) \\ &= \frac{M_2 r}{f} + I_{(M_2<t_0)}\frac{M_2^2}{2f} + I_{(M_2\geq 0)}\frac{1}{2f}\left[M_2^2 - (M_2-t_0)^2\right] \\ &\quad + I_{(t_0\leq M_2<f-r)}\frac{1}{2f}(M_2-t_0)^2 \\ &\quad + I_{(M_2\geq f-r)}\frac{1}{2f}\left[(M_2-t_0)^2 - (M_2-f+r)^2\right]. \end{aligned}$$

Again, the conditions in the above equation is not mutually exclusive.

Given the values of t_0, M_1, M_2, β and γ, we can then solve $F(\beta,\gamma)$ using off-the-shelf non-linear optimization solvers. The evaluation of our solution is presented in Section 6.

5 System Design

In this section, we give an overview of our vehicular phone-to-phone communication system, and discuss in detail a few important design issues.

5.1 System Overview

Shown in Fig. 4b is our prototype system as installed in a vehicle. A close-up shot of the various hardware components used in our system is shown in Fig. 4a. The Android phone (Galaxy Nexus [32] or Nexus S [33]) is placed under the windshield of the vehicle and is connected to the car charger.[1] The prototype application running on the phones collects and shares various driving data, including GPS trajectories, car engine OBD (onboard diagnostic) readings [34], as well as motion (accelerometer and gyroscope) data traces. The collecting and sharing of such location, car engine and motion data exemplifies a participatory sensing app that has a focus on how people's driving patterns and habits affect their vehicles' fuel consumptions. Our prototype system operates in completely autonomous manners, needing no human intervention.

The GPS, accelerometer, and gyroscope data traces are collected from the phone's corresponding built-in sensors. The engine OBD data is read using the ELM 327 OBD-to-bluetooth adapter plugged into the car's OBD-II port, and then transmitted to the phone via bluetooth.

All collected sensory data is temporarily stored locally on the phone in a database. Whenever an available WiFi AP is detected in range, data is offloaded to the back-end server and then deleted from the phone's local storage. When two vehicles are i) within the communication range of each other, and ii) in compatible modes (i.e., one as a hotspot and the other client), they exchange data until they move out of each other's communication range or complete sharing all their stored data.

5.2 Design Issues

We next discuss several important issues in our system design: 1) Adaptive system update; 2) which transport layer protocol to use; 3) scheduling during data transfer; 4) data prioritization; 5) multi-vehicle communication policy; and 6) learning WiFi AP maps.

1. Please note that in this paper we do not consider phone's battery consumption as it can be plugged into the car charger during in-vehicle operation.

5.2.1 Adaptive System Updates

In our targeted vehicular scenarios, phones enter and leave communication ranges with each other or WiFi APs from time to time. Therefore, the parameters M_1 and M_2 are not unknown or fixed, especially when the system starts running with no available historical data. We explain here how these parameters are computed and updated dynamically as the system evolves.

We treat β, γ, M_1 and M_2 as random variables. Then the optimal values of r and s are

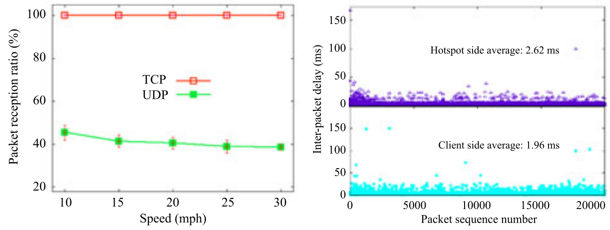
$$\langle r^*, s^* \rangle = arg \max_{r,s} E_{M_1,M_2}[E(\beta)T_1 + E(\gamma)T_2].$$

Initially the distributions of these parameters are unknown. We therefore just make an initial guess at this stage. Then as the system runs, detailed data of the parameters are sent back and the corresponding empirical distribution are updated. The parameters M_1 and M_2 represent the time durations of two cars, or a car and a WiFi AP, being within communication range of each other, respectively. Thus, they are a function of car velocity and distance, which is a known constant. The expectations of β and γ can be estimated by $N_c/(N_c + N_w)$ and $N_w/(N_c + N_w)$, where N_c is the total number of vehicle-to-vehicle meeting events and N_w vehicle-to-AP. Whenever a phone-to-phone connection or a phone-to-AP connection is established, the velocities of the vehicles are transmitted and eventually will reach the backend server. As this information accumulates, the empirical distributions of the parameters is updated. Consequently, a new $\langle r^*, s^* \rangle$ is generated and then sent back to the vehicles. As data accumulates, by the law of large number, the empirical distributions converge to the true distributions of these parameters, therefore, $\langle s^*, t^* \rangle$ will asymptotically lead to optimal system performance.

To disseminate the updated parameters into the network, we allow phones to receive this information via the cellular data channel. As the amount of data needed for this is negligible compared to other mobile sensing data, the whole network is updated immediately with only a tiny extra cost.

5.2.2 Transport Layer Protocol

TCP and UDP have their own strengths and weaknesses. To decide which one to use, we conduct a series of experiments to compare their performance in our system. During each experiment, two vehicles start at two ends of a long street, and move toward each other at fixed speeds until they reach the other end of the street. One phone serves as the hotspot and the other client. The client continuously sends data packets to the hotspot after connecting to it upon entering communication range. Packet sequence numbers are used to simulate sensory data for transmissions. TCP and UDP communications are measured separately. In addition, we optimize the TCP real-time responses to improve system efficiency by turning off the Nagle's Delay option [35], which is used to purposefully delay transmission, increasing bandwidth at the expense of latency. The packet reception ratio (PRR) under varying car speeds, ranging from 10 to 30 mph, is recorded. The experiments are repeated on different streets to minimize the effect of external noise.



(a) Packet reception ratio under varying speeds (TCP/UDP)

(b) Pattern of inter-packet latency on both hotspot and client sides.

Fig. 5. Empirical results for design decisions.

Results are shown in Fig. 5a, from which we see that UDP results in significant packet losses, only receiving about 40 percent of packets on average under all speeds. We also measure the PRR in the stationary case when two vehicles are parked near each other. We find that packet losses rarely occur, implying that the losses are mainly due to unreliable wireless links in mobile situations. Worse, we observed that the UDP packet losses occurred throughout the transmission period. Also, we measure that the inter-packet latency is around 1.5 millisecond for UDP and 1.9 millisecond for TCP under all speeds. This result indicates that the data transfer efficiency of TCP is comparable to UDP when the Nagle's Delay option is turned off, and the number of packets received using UDP is only 51 percent of the number using TCP, thus TCP results in a more efficient data transmissions. Therefore, we chose to use TCP in our system.

5.2.3 Scheduling During Data Transfer

Upon establishing connection between two phones, we can either schedule the transfers to take place in a serial manner (e.g., A sends to B, then B sends to A), or have them in parallel, i.e., two separate threads on each phone, one for sending and the other receiving, so the OS takes care of the lower-level scheduling. The former approach achieves controlled scheduling, but it is difficult to decide an appropriate time slot value, and introduces switching overhead. For the latter, we need to check whether transmission performance is affected and fairness provided. Therefore, we repeat the two-vehicle experiment with the latter approach. The inter-packet delay times on both sides is shown in Fig. 5b. We see that the data transfer with two threads is reasonably fair for both directions—the average inter-packet latency at the hotspot side is 2.62 ms, only slightly higher than the 1.96 ms measured at the other side. Considering that the inter-packet delay is 1.9 millisecond when transferring in one direction, using two separate threads can improve the transmission efficiency by 69.5 percent. Thus we choose this approach in our system design.

5.2.4 Data Prioritization

A data prioritization algorithm is designed to handle the undeterministic meeting time problem in participatory sensing applications. This algorithm enhances system performance by assigning different priorities to generated sensory data, such that representative data are transmitted first, followed by more detailed data. Hence, if data transfer is interrupted before all data can be transmitted, the most informative data will have been transmitted for the given transfer size.

The importance of data can be assessed differently under different application scenarios. In the absence of an application-specific assessment, many generic approaches have been proposed, such as FIFO, Latest-first, and Random. However, we argue that a different metric should be used in mobile participatory sensing applications, because data objects have affinity to the physical world, in that they are related to state of the world at a given location and time. To maximize coverage of the physical world to which the data pertains, it is therefore best to increase the "diversity" of transmitted data, which is a sampling problem. By maximizing diversity, the receiver gets the "big picture" quicker, as opposed to delivering lots of pieces of some content, and none or little of other content. The idea of maximizing diversity of unstructured data (such as images) was recently investigated in PhotoNet. While PhotoNet explicitly considered pictorial data, we evaluate our solution using other types of sensory data from a real testbed and compare it with conventional solutions including the timestamp-based and random exchange methods.

Our data prioritization algorithm works as follows: 1) We define semantic distance to prepresent how "different" two pieces of data are. This distance metric can be different in various participatory sensing applications. Some common distance measures include the Euclidean Distance for GPS location data and Kullback-Leibler Divergence for pictures. 2) After defining the semantic distance, we divide all sensory data into several clusters. Each cluster represents one certain physical event. Clustering itself is well-known to be NP-hard, and can be approximated by Lloyd's algorithm. Since the vehicle-to-vehicle meeting rate is typically small in delay-tolerant vehicle networks, we can run the clustering algorithm in the background on each phone, and update clusters as new sensory data are generated. 3) Finally, when two vehicles meet, each of them sends data to its peer by sampling these clusters in a round-robin way, starting with the centroid of each cluster.

5.2.5 Multi-Vehicle Communication Policy

We briefly talk about the scenario under which multiple vehicles are within communication range of each other. In our smartphone-based vehicular sensing system, let's assume they form a star-topology network, the hotspot acts as the center and other clients connect to it. The number of clients is limited by the capacity of the hotspot (e.g., the maximum number of connections for iPhone 5 is five, as confirmed by AT&T and Verizon). The hotspot communicates to its clients simultaneously via multiple threads, and data from one client flows to others through the hotspot.

One choice in the multi-vehicle scenario is to switch their roles dynamically for better global communication opportunity. For instance, a client A connects to a hotspot B, and they start transmitting data. Then A notices that another three hotspots, C, D, and E, appear in its wifi list. Hence, the best way for this local area network is to switch A as a hotspot, and the other four phones as clients, so as to get everyone involved in communication. However, this approach suffers from two main drawbacks. First, it requires an extra switch time for the client (e.g., A) to notify each hotspot to switch to client and then switch itself into a hotspot. Second, the mobility of these vehicles are undeterministic, thus it is hard to judge whether this switching process is worthy in general. Therefore, we decide not to support multi-vehicle communication in our system, as also reasoned about in Section 3.

5.2.6 WiFi AP Maps

We also recognize that having prior knowledge about WiFi AP maps could help optimize our system. Several such maps exist, being managed by the government [36] and wireless operators [37]. However, the availability of these maps is a big challenge since they are typically not made public. What's more, WiFi APs are generally designed to cover indoor environments (e.g., Cafe, office, etc) and thus are not largely accessible in vehicular settings. One possibility is to let participants' phones record local WiFi AP information and share to the central server. The server can then derive a global map and broadcast it back to all participants. This approach, however, is problematic as different participants may have different accesses to different APs. This may lead to inaccurate estimations. In addition, the highly mobile vehicular environment can lead to unstable communication patterns and subsequent conflicting results on the central server. We thus decide not to assume the availability of WiFi AP map information in this paper.

6 Evaluation

After describing our analytical model and discussed system design details, in this section we evaluate the performance of our automatic phone-to-phone communication scheme for vehicular networking applications. We report findings from our campus-wide deployment, and present optimization results through simulation experiments using a larger-scale real-world taxicab dataset.

6.1 Experiment Setup

We conducted a human subject study.[2] 35 people participated (university faculty, staff, and students of both genders, ranging from early 20 s into late 40 s, from various departments) averaging 2 weeks each, and collectively drove for around 4,000 miles. While we expect a mobile sensing application to run on participants' own phones, in our study we gave people phones pre-loaded with our test application. A mixture of both Galaxy Nexus and Nexus S phones were given to participants to be installed in their own vehicles. No specific driving routes were pre-selected; all participants were asked to drive normally and carry out their daily routines as usual. Comprehensive logging information was displayed on the phone during the running of the system, as illustrated in Fig. 4c, to notify the participants of the status of the system if they were interested.

TCP communication is used with Nagle's Delay disabled, as we learn from our prior tests that having this option enabled has negative impact on communication

2. The study was conducted under IRB protocol #10092.

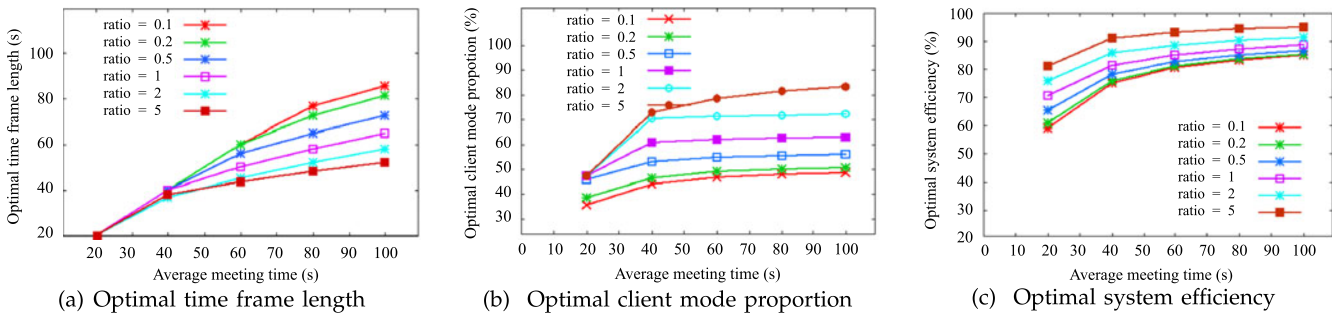


Fig. 6. Results under varying vehicle-AP/vehicle-vehicle meeting time durations and ratios.

throughput. The switching overhead is estimated to be at about 3.5 seconds for Galaxy Nexus and 6.9 seconds for Nexus S phones. During the data transfer process, two separate threads are spawn concurrently, one for sending and the other receiving. Fifty consecutive data samples are combined into one larger packet before sending in order to improve throughput.

6.2 Experiment Results

6.2.1 Analysis

We estimate the values of t_0, M_1, M_2, β and γ from data collected in our deployment, and investigate the relationship between optimal parameters and various system coefficients, under varying average meeting time durations (M_1, M_2) and vehicle-AP to vehicle-vehicle meeting ratios $(\frac{\gamma}{\beta})$.

Fig. 6a shows how optimal time frame lengths $(2t_0 + r + s)$ are affected by meeting times and ratios. We see that, when the average meeting time is below 40 s, meeting ratios have little effect on optimal time frame lengths. When the average meeting time increases beyond 40 s, the optimal frame lengths differ considerably as the meeting ratio varies—the more dominant vehicle-AP meetings are (as opposed to vehicle-vehicle meetings), the shorter the optimal frame lengths become. We also notice that as the average meeting time increases beyond 40s, the growth of the optimal frame length slows down.

Fig. 6b illustrates how optimal client mode proportion $(\frac{s}{2t_0+r+s})$ changes with different meeting times and ratios. We observe that, as the vehicle-vehicle meeting ratio decreases, the optimal client proportion increases. In particular, when vehicle-vehicle meetings are about 10 times that of vehicle-AP ones, the hotspot and client proportions are roughly the same with each other; On the other hand, the optimal client mode proportion increases beyond 80 percent when vehicle-AP meetings become dominant. These results suggest the following, i) In a dense vehicular network, in order to achieve the highest system efficiency, phones should spend approximately the same amount of time in hotspot and client modes; and ii) In a sparse vehicular network, phones should stay in client mode as much as possible in order to maximize the probability of communicating with WiFi APs. We can also easily see from the figure that the optimal client mode proportion increases when the average meeting time lengthens.

Fig. 6c shows the optimal estimated system efficiency $(\beta\frac{T_1(r^*,s^*)}{M_1} + \gamma\frac{T_2(r^*,s^*)}{M_2}$, note that T_1 and T_2 are functions that take the optimal parameters r^*, s^* as inputs) varies with meeting times and ratios. We see that the efficiency increases monotonically with both the average meeting time and vehicle-AP meeting ratio.It is quite promising that data communication takes up over 55 percent of meeting times in almost all cases, and even reaching above 90 percent in certain cases (higher vehicle-AP meeting ratio and long meeting time).

6.2.2 Deployment Results

First of all, we observe large variations in the frequencies of offloading events (where data are sent to the backend server via WiFi APs) among different users. Many users rarely move under WiFi AP coverage areas during the deployment period. As shown in Fig. 7, 64 percent of users meet access points less than once per day on average, and only 8 percent of users can offload data more than three times per day. We also notice that there are 28 percent of users who offloaded data only once or not at all during the entire deployment study. These results indicate that the end-to-end delay time for uploading sensory data can be very high due to the fact that open wireless access points are not widespread. This greatly reduces performance in participatory sensing applications where eventual data upload is expected within a short period of time.

Next, we observe that 13 out of the 25 participants met others at least once during the deployment. Several users (e.g., User 1, 5, 7, 16, and 20) were involved in vehicle-to-vehicle communications at least twice. Some users received a lot more data from peers than others. For instance, User 5 obtained 11.2 megabytes of sensory data and User 20 got 8.6 megabytes. We found in some meeting events that no data were actually exchanged. There are two possible reasons for this; the hotspot is just about to switch to the peer mode, or they ran outside the range of each other. After removing this type of event, we sort out eight total encounter events where sensory data sharing actually took place. This may seem little, but is in fact significant considering we had only 25 users and their routes spanned 2.7 square miles. In a

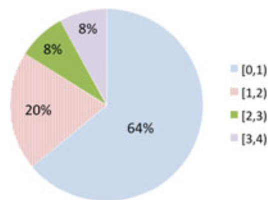


Fig. 7. Percentages of offloading events per day.

TABLE 2
More Information on Sharing Events (Event 1-4 Involve Data Sharing between a Galaxy Nexus Phone and a Nexus S Phone, whereas Event 5-8 Two Nexus S Phones)

Event	Peers	Duration (s)	Throughput (kbps)
1	1,5	139	71.9
2	1,5	136	178.1
3	1,15	219	324.8
4	1,5	477	147.6
5	20,21	1.1	610.5
6	20,1	5.4	136.3
7	12,14	0.8	746.5
8	23,24	5.4	7.3

deployed participatory sensing service, the odds of encounters grow quadratically in the number of participants, and the number of participants will likely be at least an order of magnitude or two higher than that of our small scale study.

Information about encounter events is summarized in Table 2. Among them, four events were between a Galaxy Nexus phone and a Nexus S phone, and the other four were between two Nexus S phones. Four events lasted longer than two minutes, while the other four were very short, ranging from 1 second to 5 seconds. As we looked into the GPS data, three of these four short durations occured because they finished exchanging their sensory data in local databases and no new data were generated, and one because they drove outside the communication range of each other.

We also calculate the total throughput in these eight sharing events. As shown in Table 2, the throughput achieved was as high as 746.5 kbps. Additionally, large variations in throughputs is observed, again. One important reason is that, after the exchange of sensory data from local databases is finished, new sensor readings continue to be generated continuously. The connection is kept and each phone waits for the next packet containing 50 samples from the other phone. This system behavior significantly expands the duration of data sharing and reduces the actual throughput, but in fact most time is spent on waiting for new packets with idle communication channels.

Fig. 8 presents the amount of offloaded data for those participants who received data from others. Data offloaded for oneself and peers are recorded separately. We can see

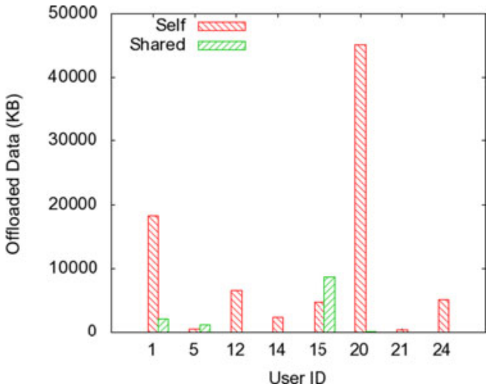


Fig. 8. Data offloaded for self and others.

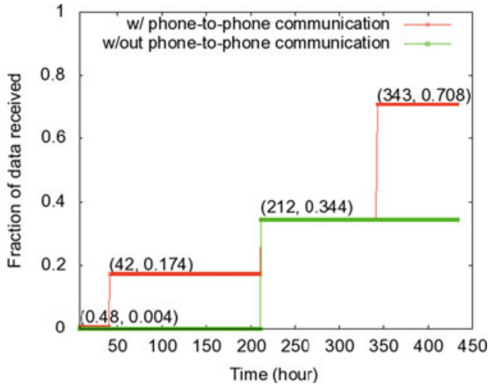


Fig. 9. Replay of fractions of data received at server side for User 5 with and without phone-to-phone communications.

that two users offloaded more shared data than self-generated data. For instance, User 5 offloaded 581 kilobytes of data generated by itself and 1,231 kilobytes of shared data. Similarly, User 15 offloaded 4,700 and 8,625 kilobytes of self-generated and shared data, respectively. These results show that vehicle-to-vehicle communication can help improve the packet reception ratio within the network.

To further investigate the benefits brought about by sharing data among peers, we inspected the sensory data reception process at the server side for several users after the deployment. For instance, Fig. 9 shows the fraction of all generated data that are delivered to the server at different time within the deployment for User 5. The reason we select User 5 is that, this participant only offloaded data once in 18 days, and shared data with peers three times. If the system were to run without phone-to-phone communication, only 34.4 percent of the data would be delivered at the hour of 212 approximately. However, with the sharing scheme enabled, the server got 0.44 percent of data 30 minutes after the deployment started. This value increased to 17.4 percent at point 42 when more data are offloaded by another participant. For these 17.4 percent of data, the end-to-end delay time was reduced by 81 percent. In addition, the fraction of received data finally reached 70.8 percent because another user helped offload more data at the point of 343. Even if we assume that these extra 36.4 percent of data are offloaded at the end of deployment in the non-DTN scenario, enabling phone-to-phone communication decreases the overall average end-to-end latency by 46.1 percent, which is a significant improvement. As a result, in this case, our solution leads to a 106 percent increase in packet reception ratio and 46.1 percent decrease in delay time for participants who rarely reach access points.

Next, we take a look at the location and time information of the sharing events in our deployment study. Exploiting the temporal/spatial characteristics may increase potential benefits of vehicle-to-vehicle communication in the future, leading to more efficient data transmission protocols based on social clusters, and specific rules in hotspot switching models, that allow staying longer in the peer mode or the hotspot mode depending on how likely it is to meet other peers or access points based on historical data.

Fig. 10 shows the locations of the sharing events listed in Table 2. We can see that the eight sharing events took place at five different locations. Three of the locations are parking

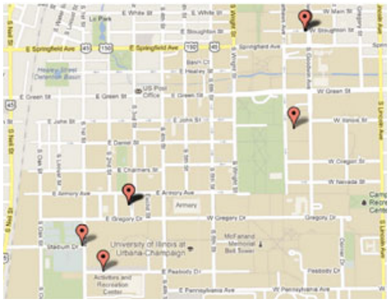


Fig. 10. Locations for sharing events.

TABLE 3
Times of Sharing Events

Event	Day of Week	Time of Day
1	Thursday	16:27
2	Saturday	10:23
3	Friday	22:52
4	Sunday	22:07
5	Wednesday	15:09
6	Saturday	17:39
7	Tuesday	17:45
8	Wednesday	15:24

lots, one at a stop sign, and one on a street. Among the three parking lot cases, two are near department buildings on campus, and one is in a residential area. Three sharing events occurred in the residential area's parking lot, possibly indicating that some users may live close to each other, or they have common friends in the same area. Two events occurred in a parking lot near department buildings, which implies that several users may work in the same or nearby departments, and they sometimes drive to work around the same time.

We also recorded the timestamps of all data sharing events, as shown in Table 3. It is easy to see that six different days in a week appear in the table, which means the sharing events widely spread weekdays and weekends. In addition, we observe that five events are in the afternoon, two in the evening, and only one in the morning.

6.2.3 Data Prioritization

We proceed to evaluate the diversity-based data prioritization algorithm, taking advantage of the sensory data we collected during deployment. Our algorithm is compared with two conventional methods, timestamp-based (oldest-first) and random, under the same experimental settings.

Each piece of data in our deployment includes five portions: 1) the node ID that generates this data (indicated by phone IMEI, note that this may be a shared data and thus a different IMEI), 2) timestamp, 3) GPS location, 4) driving status (i.e., speed, acceleration, and gyro), and 5) OBD-II data. Each data sample is 207 bytes long. In our implementation of semantic distance between sensory data, we use the timestamp and nodeID as a combinational binary decider. If the time gap between two data points is beyond some threshold (30 minutes in our experiment), or the node IDs are different, we assume they belong to different clusters and assign a large distance value to them. Otherwise, the Euclidean distance of GPS locations is calculated as the semantic distance between the two data items. In this way, sensory data are classified into 907 clusters and transmitted based on these clusters in a round-robin fashion.

During our experiment, two vehicles start at two ends of a long street initially, and run facing each other at fixed speed until they reach the other end of the street. Each of them has a Galaxy Nexus S phone placed under the windshield. Each phone has 20,000 pieces of continuous data in local storage. One of them is set as the hotspot and the other peer, and they start exchanging data with TCP when entering the communication range of each another, using the three candidate prioritization algorithms separately. The timespan of these 20,000 data is equally divided into 1,000 regions, and the fraction of regions covered by received data is recorded. The experiments are repeated to eliminate the effect of noise.

Fig. 11 shows the region coverage under varying driving speeds, with our proposed diversity-based approach, random transmission, and timestamp-based oldest-first approach. We can see that the region coverage slightly decreases as the speed increases for all candidates, this is reasonable since the number of received packets drops gradually. Additionally, when the speed varies from 15 to 30 mph, the coverage of diversity-based approach decreases from 96 to 85 percent, while the random approach drops from 74 to 63 percent and the timestamp-based approach from 22 to 13 percent, respectively. The average coverage for three candidate algorithms are 89.8, 70.5, and 17.5 percent, respectively. This is mainly because both the random and timestamp-based methods are not designed to reach as many regions as possible. These results demonstrate that the diversity-based method outperforms the other two candidates to achieve the highest region coverage.

6.3 Larger-Scale Simulation Results

Our deployment and human subject study help us get initial ideas of how our proposed system behaves. To analyze the system performance in a much larger scale, we turn to simulation experiments using the T-drive real-world taxicab dataset [29], [30], which contains the GPS trajectories of 10,357 taxicabs during the period of February 2nd through February 8th, 2008 in Beijing. To better represent our mobile sensing application scenario, we select the central part of city and filter out the suburb area where vehicles are sparse.

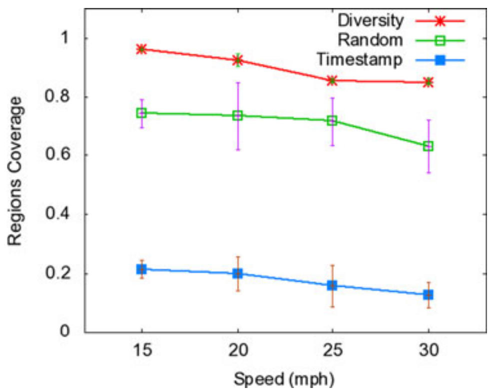
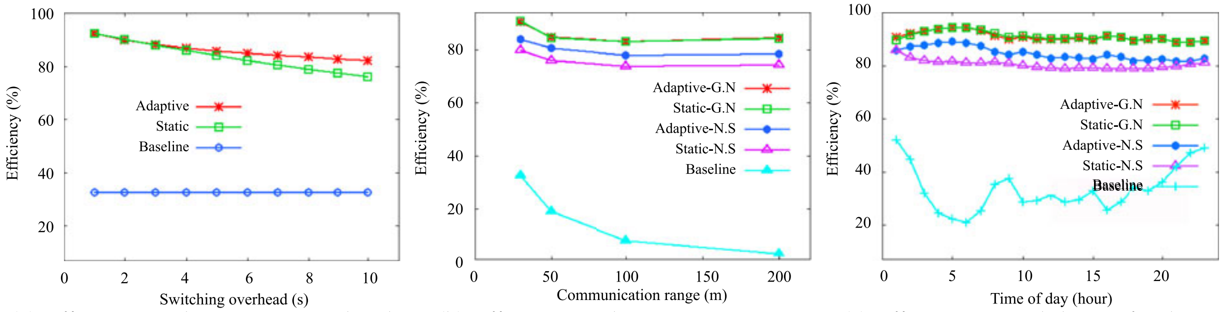


Fig. 11. Region coverage under varying driving speeds.



(a) Efficiency under varying overhead (b) Efficiency under varying range (c) Efficiency in each hour of a day

Fig. 12. Experimental results of system efficiency.

Thus our experiments contain 9,211 taxicabs, covering the central Beijing area. We focus on evaluating the system efficiency of our proposed optimization approach in this set of larger-scale simulation experiments.

We assume that 10 percent of this area is covered by WiFi APs to measure the performance of offloading events. This number is motivated by results from other large cities such as San Francisco and Seattle [38]. These WiFi APs are spread out equally in the central part of the area.

The communication range of WiFi APs and taxicabs are set to 250 and 30 meters, respectively. Both are based on our actual measurements. We also experiment with the situations where the taxicabs' communication range varies from 30m to 50 m, 100 m, and 200 m, in order to investigate the cases where the next generation phones are more powerful and capable of achieving larger communication ranges. Other system settings and parameters, including data generation and offload process, follow that of our small-scale deployment study.

We carry out the simulation using the T-drive dataset as follows. For the first 24-hour's data, we first extract meeting intervals by recording all vehicle-vehicle and vehicle-AP pairs that are in communication range at each time point, then compute the optimal parameters based on the analytical model discussed in Section 4. Finally we apply these parameters to the meeting intervals and calculate the overall system efficiency under three different candidate approaches: *Adaptive*, *Static*, and *Baseline*. *Adaptive* updates system parameters every hour based on historical data and applies them to all vehicles in the network. *Static* only uses the data from the first hour to calculate the optimal parameters, and then remains the same during the whole process. *Baseline* considers the baseline case in which phone-to-phone communication is not enabled.

We first investigate how system efficiency changes as the switching overhead (t_0) varies. Fig. 12a shows the results with the mode switching overhead ranging from 1s to 10 s, where phones' communication range is set to be 30 m. We see that the system efficiency for *Adaptive* performs slightly better than that of *Static* and is over 80 percent for all cases, specifically, 90 and 90 percent for Nexus 4 ($t_0 = 2.1$ seconds), 88 and 86 percent for Galaxy Nexus ($t_0 = 3.5$ seconds), and 84 and 79 percent for Nexus S ($t_0 = 6.9$ seconds). This indicates that our proposed solution can achieve high system efficiency using off-the-shelf smartphones and thus is highly practical. Also, since the *Baseline* approach does not allow phone-phone communication functionality at all, the system efficiency remains at 33 percent, which is just the ratio of overall phone-AP to all meeting time.

Fig. 12b shows the system efficiency under varying phone communication ranges when the mode switching overhead is 3.5 and 6.9 seconds, to emulate the use of Galaxy Nexus (G.N) and Nexus S (N.S) phones. We see that the efficiency of both *Adaptive* and *Static* does not change much as the transmission range increases. The efficiency of *Baseline* decreases as transmission range goes up. We also notice that, again, the system efficiency for *Adaptive* performs only slightly better than *Static* for both phones. This suggests that in a relatively dense vehicular network setting, our proposed solution quickly converges to optimal system parameters and does not need extensive training phase.

We next study the time-of-day system efficiency in an hour-by-hour fashion. As Fig. 12c shows, the system efficiency measurements for both *Adaptive* and *Static* do not change much throughout the day, implying that both approaches work quite well consistently. On the other hand, we see that the *Baseline* approach leads to large oscillations, mainly due to the shift of traffic patterns throughout the day, with a higher vehicle-AP meeting ratio in the evening. Therefore, our proposed approach, be it *Adaptive* or *Static*, delivers a much more stable and predictable system performance than the baseline.

Finally, Fig. 13 shows the application-level benefit that direct phone-to-phone communication brings about. We record the delay time of delivery for data generated in the first hour by all 9,211 taxicabs, with and without our solution and under varying transmission ranges. The throughput of phone-to-phone communication bandwidth is set to 746.5 kbps, obtained from measurements in our deployment. As we can see, enabling phone-to-phone communication largely decreases the delay time of data delivery, by

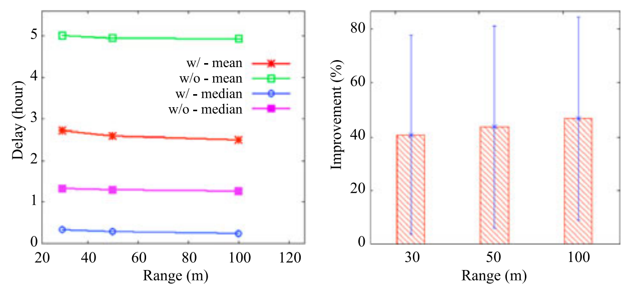


Fig. 13. Improvement of delay time with phone-to-phone communication.

more than 40 percent and up to about 50 percent on average for all communication ranges tested. More concretely, our solution helps reduce the average delay time from 5.0 to 2.7 hours, and the median from 1.3 to 0.3 hour. In addition, as the transmission range increases, the improvement by our solution also increases because it leads to more occurrences of data transfers among taxicabs. These results indicate that direct phone-to-phone communication significantly improves data collection and sharing in vehicular networking applications.

7 Conclusion

In this paper, we present the design, implementation, and evaluation of a novel optimized vehicular mobile system that leverages both phone-to-phone and phone-to-AP communications from vehicle-resident smartphones. Our proposed solution optimizes vehicle meeting communication efficiency, does not require any change to existing infrastructure, and is completely transparent to end users. Results from our 35-vehicle two-month campus-wide deployment and a large-scale real-world dataset simulation demonstrate that our approach significantly reduces data transfer delay time and maintains over 80 percent (90 percent in certain cases) system efficiency. Given the popularity of smartphones and importance of vehicular networks, we believe that this work will motivate further research on leveraging human encounters in mobile sensing and networking applications.

Acknowledgments

This work was partially supported by NSFC-GG2260080042, NSFC-BJ2260080039, and the USTC Innovation funding DG2260010011. X. Sun and S. Hu are co-first authors and contribute equally to the paper.

References

[1] R. Yanggratoke, A. Azfar, M. J. P. Marval, and S. Ahmed, "Delay tolerant network on android phones: Implementation issues and performance measurements," *J. Commun.*, vol. 6, no. 6, pp. 477–484, 2011.

[2] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding, "Cool-tether: Energy efficient on-the-fly wifi hot-spots using mobile phones," in *Proc. 5th Int. Conf. Emerging Netw. Experiments Techn.*, 2009, pp. 109–120.

[3] N. Thompson, S. C. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring replicants: Congestion control for intermittently-connected networks," in *Proc. IEEE InfoCom*, 2010, pp. 1–9.

[4] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proc. IEEE InfoCom*, 2009, pp. 846–854.

[5] N. Banerjee, M. D. Corner, and B. N. Levine, "Design and field experimentation of an energy-efficient architecture for DTN throwboxes," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 554–567, 2010.

[6] K. W. Chen, "Cafnet: A carry-and-forward delay-tolerant network," Ph.D. dissertation, MIT, Cambridge, MA, 2007.

[7] R. Crepaldi, R. Beavers, B. Ehrat, M. Jaeger, S. Biersteker, and R. Kravets, "Loadingzones: Leveraging street parking to enable vehicular internet access," in *Proc. 7th ACM Int. Workshop Challenged Netw.*, 2012, pp. 23–30.

[8] R. Crepaldi, M. Bakht, and R. Kravets, "Quicksilver: Application-driven inter-and intra-cluster communication in vanets," in *Proc. 3rd ACM Int. Workshop Mobile Opportunistic Netw.*, 2012, pp. 69–76.

[9] J. Zhao and G. Cao, "Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1910–1922, May 2008.

[10] IDC. (2013). Worldwide smartphone market expected to grow 55% in 2011 and approach shipments of one billion in 2015, according to IDC [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS22871611

[11] 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 *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 337–350.

[12] J. Yang, S. Sidhom, G. Chandrasekharan, T. Vu, H. Liu, Y. C. abd M. Gruteser, and R. Martin, "Detecting driver phone use levering car speakers," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 97–108.

[13] R. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. Abdelzaher, "GreenGPS: A participatory sensing fuel-efficient maps application," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 151–164.

[14] 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 *Proc. 8th Int. Conf. Mobile Syst., Appl. Serv.*, 2010, pp. 123–136.

[15] R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Serv.*, 2011, pp. 99–112.

[16] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 323–336.

[17] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009, pp. 85–98.

[18] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," in *Proc. 4th Int. Conf. Embedded Netw. Sens. Syst.*, 2006, pp. 125–138.

[19] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 173–184.

[20] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han, "Privacy-aware regression modeling of participatory sensing data," in *Proc. 8th ACM Conf. Embedded Netw. Sens. Syst.*, 2010, pp. 99–112.

[21] H. Ahmadi, T. Abdelzaher, J. Han, R. Ganti, and N. Pham, "On reliable modeling of open cyber-physical systems and its application to green transportation," in *Proc. IEEE Int. Conf. Cyber-Physical Syst.*, 2011, pp. 87–89.

[22] (2013). Customers Angered as iPhones Overload AT&T. NY Times, September 2009 [Online]. Available: http://goo.gl/KtuInV

[23] (2013). iPhone overload: Dutch T-Mobile issues refund after 3G issues. Ars Technica, July 2010 [Online]. Available: http://goo.gl/6DCNYh

[24] H. Liu, S. Hu, W. Zheng, Z. Xie, S. Wang, P. Hui, and T. Abdelzaher, "Efficient 3G budget utilization in mobile participatory sensing applications," in *Proc. IEEE InfoCom*, 2013, pp. 1411–1419.

[25] (2013). The Code for the New wpa_supplicant [Online]. Available: https://github.com/kelvie/android-wpa_supplicant

[26] C. Foresman, "Wi-fi direct protocol to ease peer-to-peer wifi connections," *Ars Technica*, 2009, http://arstechnica.com/gadgets/2009/10/wi-fi-direct-protocol-to-ease-peer-to-peer-wifi-connections

[27] (2013). Android issue 82: Support ad hoc networking [Online]. Available: http://code.google.com/p/android/issues/detail?id=82

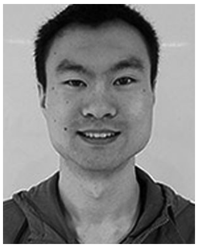
[28] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "Wifi-opp: Ad-hoc-less opportunistic networking," in *Proc. 6th ACM Workshop Challenged Netw.*, 2011, pp. 37–42.

[29] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 316–324.

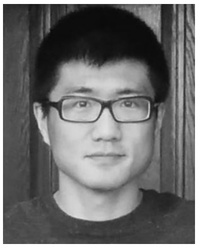
[30] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2010, pp. 99–108.

[31] S. Nirjon, A. Nicoara, C.-H. Hsu, J. Singh, and J. Stankovic, "Multinets: Policy oriented real-time switching of wireless interfaces on mobile devices," in *Proc. IEEE 18th Real-Time Embedded Technol. Appl. Symp.*, 2012, pp. 251–260.

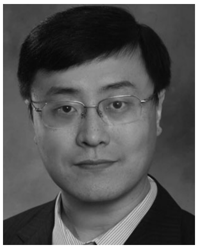
[32] (2013). Galaxy Nexus [Online]. Available: http://goo.gl/zdd9f
[33] (2013). Nexus S [Online]. Available: http://www.google.com/phone/detail/nexus-s
[34] R. Birnbaum and J. Truglia, *Getting to Know OBD II*, AST Training, ISBN 0-9706711-0-5, New York, 2000.
[35] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, New York, NY, USA: Elsevier, 2007.
[36] (2013). Washington, DC Wi-Fi Hot Spot Map [Online]. Available: http://goo.gl/cqR8KA
[37] (2013). AT&T Wi-Fi Hot Spot locations [Online]. Available: http://goo.gl/DaOf99
[38] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using wifi," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 209–222.



Xiaoshan Sun received the BS degree from USTC in 2015. He is now a graduate student in the Professor Hengchang Liu's group, Department of Computer Science and Technology, USTC. His research interests mainly include cyber physical system and data mining.



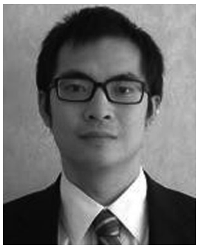
Shaohan Hu received the MS degree from the Department of Computer Science, Dartmouth College, and is working toward the PhD degree and is research assistant in the Department of Computer Science, University of Illinois, Urbana-Champaign. His main research interests lie in designing and building mobile sensing and computing systems.



Lu Su received the PhD degree in computer science and the MS degree in statistics, both from the University of Illinois, Urbana-Champaign, in 2013 and 2012, respectively. He is an assistant professor in the Department of Computer Science and Engineering, SUNY Buffalo. His research focuses on the general areas of cyber-physical systems, wireless and sensor networks, and mobile computing. He has also worked at the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is a member of the ACM and IEEE.

Tarek F. Abdelzaher received the PhD degree from the University of Michigan, Ann Arbor, in 1999, under Professor Kang Shin. He was an assistant professor with the University of Virginia from August 1999 to August 2005. He then joined the University of Illinois, Urbana-Champaign, as an associate professor with tenure, where he became full professor in 2011. His interests lie primarily in systems, including operating systems, networking, sensor networks, distributed systems, and embedded realtime systems. He is especially interested in developing theory, architectural support, and computing abstractions for predictability in software systems, motivated by the increasing software complexity and the growing sources of nondeterminism. Applications range from sensor networks to large-scale server farms, and from transportation systems to medicine.

Pan Hui received the PhD degree from the University of Cambridge, under supervision of Professor Jon Crowcroft. He is with the Department of Computer Science and Engineering, Hong Kong University of Science and Engineering, Hong Kong, and the Telekom Innovation Laboratories, Berlin, Germany. His research interest mainly includes mobile augmented reality, mobile computing, and other interesting topics related to social network.



Wei Zheng received the PhD degree from the Department of Statistics, University of Wisconsin-Madison. He is now a statistician at the Sanofi-Aventis US LLC. His research interest mainly includes statistical modeling in clinical trials, data mining in mobile systems, and various types of networks.

Hengchang Liu received the PhD degree from the University of Virginia in 2011, under supervision of Professor John Stankovic. He is currently an assistant professor at USTC. His research interest mainly includes cyber physical systems, mobile systems, named data networking, and wireless (sensor) networks.

John A. Stankovic received the PhD degree from the Brown University. He is the BP America professor in the Computer Science Department, University of Virginia. He served as chair of the department for 8 years. He won the IEEE Real-Time Systems Technical Committee's Award for Outstanding Technical Contributions and Leadership. He also won the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award (inaugural winner). His research interests are in real-time systems, distributed computing, wireless sensor networks, and cyber physical systems. He is a fellow of both the IEEE and the ACM.