Margaret Martonosi *Princeton University*

Editor: Carla Schlatter Ellis

MOBILE SENSING:
Retrospectives and Trends

It is difficult to think back to a time before smartphones existed, with their ubiquitous computing and communication capabilities, and with detailed location sensing easily available from Global Positioning Systems (GPS). In the late 1990s, when my research group began work on mobile sensing, smartphones had not yet been invented. While GPS did exist, GPS receivers were expensive, power-hungry and not widely available. Our first mobile computing project started as a power-efficiency study for a GPS-based interactive campus tour. GPS-based tour applications are familiar now, but were unheard of then, and the physical implementation was a challenge. We used a Palm Pilot PDA (personal digital assistant) connected to an external GPS receiver and an external Wi-Fi card. In those days, PDAs had neither GPS nor any wireless communication capability! Given the bulkiness of the various pieces of our "app," we carried them and their batteries around in a shoebox. Since both the GPS and the radio were quite high power (over 1W), they greatly impacted the system's battery life. Our power-efficiency work explored methods to locally cache maps on the PDA, and to power down modules when not in use.

Meeting with a biologist on campus, the conversation transitioned from the campus-tour-in-a-shoe-box to a more interesting tracking challenge: zebras in Kenya. My observation was that the state-of-the-art, fine-grained tracking system biologists wanted could be designed as an innovative sensor network built entirely from mobile tracking nodes. From that insight, the ZebraNet project began.

It has been roughly 15 years since my research began to include mobile system applications and architectures. Over that time, much has changed. Thanks to ongoing innovation driven by academic research groups and considerable industry attention, the underlying mobile technologies have matured remarkably. The applications and services have both *driven* device innovations and also benefited from them. In this retrospective, I first give a timeline summarizing my own group's research, and then use those experiences to discuss some of the future mobile trends and some lessons learned.

THE ZebraNet WILDLIFE TRACKER: 2001-07

The ZebraNet Project established the field of mobile sensor networks. An interdisciplinary collaboration between biology and computing, the goal from a biology perspective was to track wildlife movements at a fine grain in space and time, across large tracking areas (hundreds of square miles) and with *no* cellular connectivity or other installed infrastructure. In particular, certain subspecies of zebra are endangered in central Kenya, and the goal was to be able to better understand their social and migration behaviors by being able to track their movements at the granularity of minutes – not days or hours – and with location accuracy of tens of meters or less.

From an engineering perspective, no portable device at that time could achieve ZebraNet's tracking goals; building the system would require significant hardware and software innovations. In particular, at the time, Kenya had almost no rural cellular connectivity, so radioing data from an arbitrary tracking node back to a base station would require several watts of power to cover a useful range. ZebraNet instead developed energy-efficient protocols for short-range, pairwise data transfers. When two zebras (or rather, their tracking collars) discovered they were within radio range of each other, they would swap data. Eventually a zebra within range of a base station would offload all its data to the base station, including data from the many zebras with whom it might have swapped. In this way, we used store-and-forward techniques and viral routing protocols to build a delay-tolerant network (DTN) of zebras.

In addition, GPS receivers at the time were quite high power (1W), so taking position samples at the desired rate of six times per hour would quickly drain any batteries that a zebra collar could comfortably include. Our project explored options for saving GPS power, including collaborative localization [15].

With collaborative node behavior and delay-tolerant peer-to-peer data swaps as our fundamental innovations, our project also comprehensively addressed other issues including: hardware design, energy adaptation, communication protocol and software structure. Our research spanned vertically from application software layers through custom-designed middleware, hardware design, solar charging circuitry and even the physical design and implementation of the zebra collars themselves.

At the time, "Smart Dust" and sensor networks were getting attention, but focused on *fixed* sensing nodes either placed or sprinkled in the target area. A major contribution of ZebraNet was in highlighting the advantages of *mobile*

Photo, istockphoto.com

sensor nodes. Clearly, attaching the GPS-based sensors to the zebras themselves made sense for tracking their motion; other approaches based on fixed detectors or cameras would have required much more infrastructure and much higher node counts. Our peer-to-peer approach faced a fundamental chicken-and-egg problem: namely, knowing the node mobility patterns in advance would have helped inform our design decisions about the communication protocols, but as a wildlife tracking project, knowing the node mobility patterns was the desired *result* rather than input. In the end, we designed adaptive protocols we could adjust after deployment. Using mobile sensors with peer-to-peer data swaps allowed us to employ radio ranges 2-10X shorter than those of a more traditional node-to-base communication protocol. This represented 20X or more improvements in energy, allowing us to operate with fewer solar cells and smaller batteries.

ZebraNet was deployed twice in Kenya (Figure 1). It collected thousands of data points, provided biologists with never-before-seen animal behavior data and established the utility of mobile sensor networks for many problems. Some of its key ideas and innovations are quite useful today: (i) Moving sensors provide better sensing and network coverage than fixed ones, and at lower energy as well. Let the tracked entities carry the sensors. (ii) Requiring full multi-hop end-to-end route connectivity requires much greater node density than peer-to-peer techniques. DTNs with peer-to-peer swapping offer better connectivity at lower energy. (iii) For sensing across large areas, much of the energy will be in the radio. Under those circumstances, strategic on-node data preprocessing (e.g., data compression or event analysis) offers high-energy savings by dramatically reducing the amount of data actually sent off-node [12]. The ZebraNet compression work is an early example of an ongoing trend discussed in "Technology and Applications Shifts" towards on-device or near-data processing.

FIGURE 1. One collared zebra amidst a herd in Laikipia, Kenya. Photo, courtesy of Pei Zhang.

FIGURE 2. Sibren Isaacman and Zimo Zheng work with students using C-LINK.

C-LINK: 2007-09

Whereas ZebraNet was a very vertical systems research effort organized around a specific applications goal, it also led us to think more broadly about issues and opportunities for efficient, ultra-low-cost data communications. Motivated to further explore DTN applications, we turned to human contexts and deployed the C-LINK system in Nicaragua, as depicted in Figure 2 [6, 7]. C-LINK used DTN techniques similar to ZebraNet's, but this time aimed at offering low-cost Internet access in disconnected rural villages. In many places around the world, rural areas may have poor or expensive broadband connectivity, but there are many vehicles (personal cars, bush taxis or vans, postal vehicles, etc.) traveling between villages and larger cities. C-LINK used such vehicles as potential "data mules," using DTN techniques to bring web queries from the village kiosk to an Internet-connected kiosk in a larger city where data could wirelessly "jump off the bus" and be transmitted on the full Internet.

A key innovation of C-LINK was in building distributed collaborative caching layers on top of the DTN, in order to make potentially long web delays less visible to users. Collaborative caching is a software layer (running underneath a normal web browser and invisible to users after installation) that lets users choose to share their laptop's cached data with other laptops nearby. If the village is currently completely disconnected from the Internet, local web searches may still succeed due to the collaborative caching. A key challenge in collaborative caching was in designing a resilient indexing scheme that let cooperating user nodes find the data in nearby caches with little delay, and in a manner robust to nodes entering or leaving the system arbitrarily. As with ZebraNet, another challenge was in designing the system to be resilient to variations in the node mobility pattern. These recurring challenges with node mobility prediction motivated our subsequent projects discussed in "WHERE and DP-WHERE."

Some of the broader lessons learned in C-LINK include the value of a collaborative data directory that dynamically moves information from node to node, and dynamically elects new "leader" nodes as nodes enter or leave the village. Particularly in regions with severe connectivity and resource constraints, additional fault tolerance and resilience is key to a successful design. Finally, we also note the value of hybrid forms of connectivity. For example, in addition to vehicular DTNs, which are the last-resort approach for very rural areas, one might have some connectivity through weak cellular links or other methods [7]. For example, using cheap SMS messages for the "uplink" portion of a web search can cut latency by 2X, with very little cost increase, compared to an approach solely based on vehicular DTNs. Several systems now offer very fluid web-SMS hybrids similar to what C-LINK explored.

SignalGuru: 2010-11

Like ZebraNet, the SignalGuru project was also a foray into opportunistic mobile sensing, but this time vehicular rather than wildlife [8]. In today's terminology, SignalGuru was a Collaborative Intelligent Transportation System (CITS). The SignalGuru project explored methods by which cellphones and other mobile devices could collaborate to share information and solve problems, particularly those involving location-aware services. Our work implemented and evaluated a method by which dashboard-mounted smartphone cameras in moving cars could detect stoplight transitions and share red-green schedule information. This allows users within a region to collaboratively share information in order to identify "optimal" speeds at which each vehicle should travel through the traffic to reduce stops for red lights and to improve fuel efficiency.

Similar to ZebraNet, one broad lesson learned from SignalGuru was the value of *collaborative* mobile services; allowing different SignalGuru nodes to share information about traffic light transitions observed within a region was instrumental to achieving high accuracy. The node densities required to offer good performance were different depending on the traffic situation. When traffic conditions were relatively steady-state, low densities of just 1 to 2 nodes per feeder area around a stoplight were sufficient to get good performance, especially if the participating cars were on different roads into the same intersection.

Another broad lesson learned was again about the high potential value of *on-device computation*. In particular, SignalGuru demonstrated that the relatively modest compute capabilities of a smartphone at the time were sufficient to solve the real-time computer vision problem of identifying red or green traffic lights in the scene being viewed by a dashboard-mounted cellphone camera. Performing the video analysis on each local phone allowed us to greatly reduce the application's phone-cloud communication requirements; this improves both latency and energy. As noted in "Technology and Applications Shifts," this push towards on-device data processing is a strategy that is offering even higher leverage as device CPUs become more capable.

WHERE AND DP-WHERE: 2009-12

Where C-LINK and SignalGuru were about exploiting mobile sensing directly, we also pushed our research towards modeling human mobility. This was inspired by the challenges faced in ZebraNet, C-LINK and SignalGuru to build systems without a detailed design-time understanding of node mobility statistics. With a goal of supporting many different urban planning applications, our work on region-scale mobility characterizations and modeling included the WHERE [5] and DP-WHERE [11] mobility models. Performed in collaboration with AT&T, this research used coarse-grained, spatiotemporal information gleaned from gigabytes of anonymized call detail records (CDRs) to form large-scale mobility characterizations and synthetic models. On their own, these human mobility models are important for cellular communications network planning. More broadly, they have sweeping applications to many diverse challenges: controlling automobile and highway congestion, guiding the design of intelligent transport systems, estimating regional carbon footprints and even modeling the spread of contagious diseases [1, 5].

With cellular network data becoming more available, creating human mobility characterizations and models from such data might seem straightforward, but challenges arise because CDRs are spatially and temporally coarse-grained. Our CDR logs included only the cell tower location (not the user's location) and only at the time each phone call was initiated (not at any other times). In addition to being coarse-grained, CDRs convey only time and position; they do not convey whether their associated locations correspond to the user's home, work or other important places. Without such semantic information, it is difficult to abstract CDRs into parameterized models applicable to scenarios, regions or populations that vary from those for which the real-life CDR data was collected. For example, if one wants to adjust a CDR-based model to account for different telecommuting rates, one needs to know or be able to infer home and work locations. WHERE used statistical regression techniques to infer home, work and other "important places" [3] from CDR log statistics. We showed that the best techniques for identifying important places were not just based on their frequency of occurrence in the CDR trace, but also based on their duration of days in the trace and other factors. Using Markov modeling techniques on top of CDR statistics annotated with important place labels, we produced accurate models reflective of real human movement patterns in particular metropolitan regions. WHERE can compute "daily range" and other urban planning estimates at very low error, and can correctly distinguish the geographic behaviors of distinct cities such as comparing commute patterns in New York versus Los Angeles.

Finally, we also explored methods to fully privatize the data characteristics of WHERE mobility models using the technique of differential privacy. Our resulting model, DP-WHERE [11], achieves differential privacy by adding controlled noise to the set of empirical probability distributions that WHERE uses, for example, distributions of home and work locations. DP-WHERE then proceeds identically to WHERE by systematically sampling these distributions to generate synthetic CDRs containing synthetic locations and associated times. The DP-WHERE approach shows that modest revisions to a mobility model drawn from real-world and large-scale location data can allow for rigorous privacy without overly compromising its utility or accuracy. DP-WHERE was the first differentially private approach for modeling human mobility based on large sets of cellular network data. It shows that differential privacy can be achieved with only a modest and acceptable reduction in accuracy. In particular, across a wide array of experiments involving 10,000 synthetic users moving across more than 14,000 square miles, the distance between synthetic and real population density distributions for DP-WHERE differed by only 0.17 - 2.2 miles from those of WHERE. Finally, linking all the strands together, we also showed methods to use WHERE-style mobility models (based on either cell phone or census data) to inform the design of C-LINK deployments being tailored to a particular region's movement patterns.

TECHNOLOGY AND APPLICATIONS SHIFTS

The research projects discussed span more than 15 years. Those years offered a remarkable set of technology and applications disruptions; our work was both influenced by them and also a part of the wave of activities that caused them. We discuss some highlights below.

Ubiquity of radios: It is astonishing to think back to a time when mobile devices (e.g. PDAs) had no wireless connectivity and could be synced only when docked to a desktop machine. The rise and prevalence of on-device radio is matched with the prevalence of cellular and Wi-Fi base station availability to create an ambient infrastructure where it is technologically quite easy to approach 100% data connectivity across much of the world. Although radios are nearly ubiquitous, however, there are still notable costs to communication – both in terms of cellular data costs as well as in terms of time and energy. Thus, even with nearly universal connectivity, it is still worthwhile to consider concepts of near-data (on-device) computation and compression for data reduction before communicating to the cloud.

Ease of localization: Perhaps the most remarkable piece of nostalgia is to think back to a world where GPS was not only not widely available, but was actively scrambled [13]. Today, position information can be easily and accurately gathered, either by the GPSs nearly universally available on smartphones, or by other techniques that interpolate based on Wi-Fi SSID and signal strength information. In addition, work by our group and others has pointed the way for collaborative localization off of trusted nodes [15]. In today's context with location information so easily available, opportunistic mobile sensing comes nearly for free; devices are doing mobile sensing at all times. On the other hand, with that ease of localization come concerns about location privacy, which future systems will need to address at all levels, and early in the hardware and software design processes.

Transition from wimpy to capable CPUs: In early mobile and sensing devices, energy concerns tilted designers towards extremely constrained wimpy CPUs, even turning back to 8-bit processors. Today's mobile CPUs are much more capable, now including 2-4 general-purpose processors, and many specialized accelerators. The scaling of CPU capability on mobile devices is not simply the natural progression of Moore's Law, but has in fact considerably exceeded Moore's Law scaling rates, in recognition of the need for substantial compute capability on the mobile node itself [2]. Applications can now be written to aggressively exploit near-data, on-device computation, because the device will have a CPU that can handle it. Performing on-device data processing has both energy and bandwidth advantages. On typical smartphones, off-device radio energy per byte is roughly 1000X greater than on-chip CPU energy per instruction, so on-device processing can greatly reduce the off-device data communication. Such near-data processing offers huge savings in both device battery energy and downstream communication bandwidth and energy.

Application shifts: In addition to technology shifts, applications have shifted as well. The dramatic uptake of smartphones – from non-existence when we started these projects to the billions worldwide today – means that roughly one-third of the world's population has a smart sensor in their pocket at all times. This offers potent opportunities for opportunistic sensing and crowd-sourcing. Furthermore, the Big Data revolution currently underway relies fundamentally on the ability to gather many large data streams and to perform unique analyses across many of them. As such, the power and ubiquity of today's mobile nodes helps drive analytics research and application developments by making it easier and lower-cost than ever before to acquire rich, fine-grained location and sensor data.

THE "OTHER" LESSONS LEARNED

Many of the technical lessons we learned are woven into the project descriptions above. But beyond the technical, this stream of projects also taught numerous non-technical lessons as well.

Reality matters: First and foremost is the value of real deployments and at-scale datasets. Committing our projects to real deployments pulled us into research problems that other simulation-oriented researchers had not noticed or pursued. For example, in ZebraNet, we knew that once our tracking nodes had been placed onto tranquilized-then-reawakened wild zebras, there was simply no way to press a reboot button. Thus, we needed a solid plan for how to perform over-the-air updates

FROM AN ENGINEERING PERSPECTIVE, NO PORTABLE DEVICE AT THAT TIME COULD ACHIEVE ZEBRANET'S TRACKING GOALS; BUILDING THE SYSTEM WOULD REQUIRE SIGNIFICANT HARDWARE AND SOFTWARE INNOVATIONS.

on nodes out in the field. This pushed us towards a middleware layer that allowed modular, incremental, peer-to-peer updates [10]. Likewise, in our DP-WHERE work, it was important that we be able to test our differential privacy techniques on real, at-scale data. The key question was whether the blurring imposed by DP would lead to unacceptable degradation in model accuracy. Without large, realistic datasets, there is no convincing way to answer that question.

Bring your shoebox: My mobile computing research began with a project that required us to carry bits of hardware around campus in a shoebox, because no obvious integrated platform existed to perform the functions we wanted. While logistically inconvenient, I think such hassles are a necessary part of doing cutting-edge systems research and should be embraced. In systems research, pushing beyond what easily exists is the best way to get to novel points of the design space and therefore to raise the impact of the work. This was visibly apparent in our campus-tour-in-a-shoebox. In other work such as WHERE and DP- WHERE, the logistical challenges were not as visible but were very present and the logistical challenges were part of how we knew we were pushing beyond "work as usual" in the scale of data we were attacking. I try to remember (and embrace!) the mental picture of that shoebox as I embark on new research projects. If the infrastructure that exists on Day One of your research is too convenient, then you are not thinking far enough ahead.

The big issues to come: The mobile sensing work of the early 2000s needed to focus heavily on the mechanics of effectively gathering useful data, because so little device- or network-level infrastructure existed for doing that. The basics of performance and connectivity were the major focus. Now however, the infrastructure for mobile sensing has matured considerably, with much data easily, opportunistically available. So, the tables have turned. In place of challenges in getting the data, we are facing challenges in processing the data and in doing so responsibly. This means that privacy mechanisms for location data will become a foremost ingredient in future research, and should probably be placed very early in the data collection pipeline. In addition, as systems use mobility and location data more widely and in more mission-critical ways, reliability concerns also become prominent. We must be able to identify data irregularities or falsehoods before making important decisions based on them.

Overall, I entered the field of mobile systems as a newly-tenured computer architect with a background in power-aware computing, and I set out to find interesting, applied systems research problems with stringent power-efficiency challenges. Within mobile computing, I found that and more. The topic offers compelling opportunities to build interesting, at-scale prototypes and to experiment across many layers of the hardware and software stack. Looking forward, the challenge will be to take a field that has gone from niche to ubiquitous, and suffuse it with the additional rigors of privacy, security and reliability so important to future applications.

ACKNOWLEDGEMENTS

The work described here has benefited from the talents and hard work of many students (particularly Sibren Isaacman, Manos Koukoumidis, Ting Liu, Chris Sadler, and Pei Zhang) and collaborators (particularly Ramón Cáceres, Steve Lyon and Dan Rubenstein). The work was also made possible by the support of numerous NSF research grants. ■

Margaret Martonosi is the Hugh Trumbull Adams '35 Professor of Computer Science at Princeton University. Her research interests are in computer architecture and mobile computing, with particular focus on power-efficient systems. She is a Fellow of both IEEE and ACM and a recipient of numerous awards for technical achievements and mentoring. She completed her Ph.D. at Stanford University in 1993.

REFERENCES

[1] Richard A. Becker, Ramón Cáceres, Karrie Hanson, Sibren Isaacman, Ji Meng Loh, Margaret Martonosi, James Rowland, Simon Urbanek, Alexander Varshavsky, and Chris Volinsky. Human mobility characterization from cellular network data. *Commun. ACM*, 56(1):74-82, 2013.

[2] Matthew Halpern, Yuhao Zhu, and Vijay Janapa Reddi. Mobile CPU's rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction. In *HPCA*, 2016.

[3] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Stephen G. Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. Identifying important places in people's lives from cellular network data. In *Pervasive*, volume 6696 of *Lecture Notes in Computer Science*, pages 133-151. Springer, 2011.

[4] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Stephen G. Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. Ranges of human mobility in Los Angeles and New York. In *PerCom Workshops*, pages 88-93. IEEE, 2011.

[5] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky and Walter Willinger. Human mobility modeling at metropolitan scales. In *MobiSys*, pages 239-252. ACM, 2012.

[6] Sibren Isaacman and Margaret Martonosi. Potential for collaborative caching and prefetching in largely-disconnected villages. In *Wireless Networks and Systems for Developing Regions*, pages 23-30. ACM, 2008.

[7] Sibren Isaacman and Margaret Martonosi. Low-infrastructure methods to improve Internet access for mobile users in emerging regions. In *WWW (Companion Volume)*, pages 473-482. ACM, 2011.

[8] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Martonosi. SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *MobiSys*, pages 127-140. ACM, 2011.

[9] Vincent Lenders, Emmanouil Koukoumidis, Pei Zhang, and Margaret Martonosi. Location-based trust for mobile user-generated content: applications, challenges and implementations. In *HotMobile*, pages 60-64. ACM, 2008.

[10] Ting Liu, Christopher M. Sadler, Pei Zhang and Margaret Martonosi. Implementing software on resource-constrained mobile sensors: Experiences with Impala and ZebraNet. In *MobiSys*. USENIX, 2004.

[11] Darakhshan J. Mir, Sibren Isaacman, Ramón Cáceres, Margaret Martonosi, and Rebecca N. Wright. DP-WHERE: differentially private modeling of human mobility. In *BigData Conference*, pages 580-588. IEEE, 2013.

[12] Christopher M. Sadler and Margaret Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *SenSys*, pages 265-278. ACM, 2006.

[13] Mark Sullivan. A brief history of GPS. *PC World*, August 2012.

[14] Pei Zhang and Margaret Martonosi. Energy adaptation techniques to optimize data delivery in store-and-forward sensor networks. In SenSys, pages 405-406. ACM, 2006.

[15] Pei Zhang and Margaret Martonosi. LOCALE: collaborative localization estimation for sparse mobile sensor networks. In *IPSN*, pages 195-206. IEEE Computer Society, 2008.