New Directions: The Future of Sensing is Batteryless, Intermittent, and Awesome

Josiah Hester
Northwestern University
josiah@northwestern.edu

Jacob Sorber
Clemson University
jsorber@clemson.edu

ABSTRACT
Sensing has been obsessed with delivering on the “smart dust” vision outlined decades ago, where trillions of tiny invisible computers support daily life, infrastructure, and humanity in general. Batteries are the single greatest threat to this vision of a sustainable Internet of Things. They are expensive, bulky, hazardous, and wear out after a few years (even rechargeables). Replacing and disposing of billions or trillions of dead batteries per year would be expensive and irresponsible. By leaving the batteries behind and surviving off energy harvested from the environment, tiny intermittently powered computers can monitor objects in hard to reach places maintenance free for decades. The intermittent execution, constrained compute and energy resources, and unreliability of these devices creates new challenges for the sensing and embedded systems community. However, the rewards and potential impact across many fields are worth it, enabling currently impractical applications in health services and patient care, commercial and consumer applications, wildlife conservation, industrial and infrastructure management, even space exploration. This paper highlights major research questions and establishes new directions for the community to embrace and investigate.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Architectures; • Human-centered computing → Ubiquitous and mobile computing systems and tools;

KEYWORDS
Sensing, Batteryless, Intermittent, Energy harvesting

ACM Reference format:
DOI: 10.1145/3131672.3131699

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

SenSys ’17, Delft, Netherlands
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-5459-2/17/11...$15.00
DOI: 10.1145/3131672.3131699

Figure 1: A typical intermittent execution pattern for a batteryless sensor. Energy availability depends on environmental conditions and sensor behavior. Intermittent execution is a side-effect of this batteryless model, with outages of unknown length separating bursts of execution.

1 INTRODUCTION
We have a battery problem. For more than a decade, the wireless sensing community has worked together to enable long-term, affordable, sustainable sensing across many application domains, ranging from wildlife tracking to infrastructure monitoring, the built environment, wearables, autonomous vehicles, and even space exploration. We have often renamed and refined this shared vision—from “ubiquitous sensing,” to “smart dust,” the “TerraSwarm,” and the “Internet-of-Things”—and we have made progress in many areas, but sustainable long-term sensing remains elusive because our systems still depend on batteries and reliable power.

Batteries are not only convenient and ubiquitous; they are also incompatible with a sustainable Internet-of-Things (IoT). Batteries are expensive, bulky, and hazardous. They are sensitive to temperature, dangerous when not carefully protected, and short-lived—they wear out after a few years (even rechargeables). Replacing and disposing of billions or trillions of dead batteries per year will be costly and irresponsible. We can wait for batteries to improve, but we see no evidence that batteries will overcome these challenges in the foreseeable future; instead, we can leave batteries behind and design systems that depend on harvested environmental energy, store small amounts of energy in capacitors, and operate effectively even in the face of frequent power outages. With these tiny intermittently powered computers, we can gather data in hard to reach places, maintenance free, for decades. Batteryless sensing will revolutionize computing, but first we must learn how to build, program, debug, and network this new class of device.

Batteryless intermittent computing is fundamentally different from conventional computing, because intermittent devices violate one of the most basic assumptions of computing—a stable power supply. Energy harvesting is inconsistent, energy storage is scarce, power failures are inevitable, and execution is intermittent, as shown in Figure 1.
The intermittent programmer’s job is to combine fragmented variable-length execution opportunities into useful work. Batteryless, energy harvesting sensing presents critical challenges not encountered by tethered or battery-powered sensing. Firstly, in unpredictable environments, supply voltages vary and power failures can occur at any time, and failures can last seconds, minutes, or hours. Secondly, a device’s behavior affects the amount of energy it can harvest—meaning small changes in tasks can drastically change harvester efficiency. Thirdly, the programming interfaces of batteryless device are ill-defined and non-intuitive; most developers have trouble anticipating the problems inherent with an intermittent power supply.

Our goal is to make batteryless sensing practical and usable by system designers across application domains. Researchers can explore and develop new hardware designs, software techniques, tools, and programming abstractions so that developers can easily design, debug, and deploy sophisticated batteryless applications that work in spite of frequent power failures. This paper argues for a vision of the future of sensing. In this future devices are small, cheap, and are useful for the entire lifetime of the things they monitor. These devices will be easy to program, straightforward to test, and will be deployed confidently. This future must focus on relieving the cognitive burden on programmers to the point where even amateurs can confidently program, test, and deploy batteryless sensors. In the rest of this paper we discuss these new directions in the context of three application domains that are especially motivating: sensing in the built environment, on-body sensing and wearables, and large scale infrastructure and environmental monitoring.

2 BATTERYLESS SENSING: PROMISE AND CHALLENGE

The lack of a reliable energy source makes developing sophisticated applications on batteryless devices difficult. Batteryless sensors can lose power multiple times a second, and power failures can be as short as a few milliseconds, or as long as many hours or days. This intermittent operation creates problems. These problems transcend each of the new directions described in this paper, and are general to all batteryless sensors.

Constrained Resources: Batteryless sensors must scavenge energy to survive, the less energy the sensor needs to compute, the more it can execute. This necessitates devices with little memory and a little compute power like the WISP [17] or EnHants [14]. This also means that high powered, highly accurate sensors like some accelerometers, or large, bulky components like GPS are difficult to equip on batteryless sensors. Long running computations will be interrupted because of the high cost on under-powered hardware. Techniques like Mementos [16] and others [1, 3, 13, 19] attempt to conserve resources by checkpointing computation. Dewdrop attempts to schedule tasks more intelligently [4], while QuarkOS [20] breaks tasks into micro tasks. Others approaches change the hardware to try and mask the intermittent operation [11, 15] All of these approaches have enabled better, more efficient computation on energy harvesting devices, but challenges remain.

Uncertain Environments: Batteryless sensors are intended to be deployed in a huge variety of applications—therefore scavenging from a wide variety of energy mediums. Wireless energy transfer adds more possibilities, now energy is potentially infinite, yet intermittent. Harvester and wireless transfer research is increasing [2], enabling energy harvesting devices that can harvest more energy at smaller size. Each type of harvesting—from thermal to kinetic, solar, Wifi, and RFID—have different behaviors, different volumes of energy, and different schedules (if any). Solar energy is unavailable at night, RFID and Wifi are rarely useful outside of a built environment, and thermal harvesting usually requires large temperature gradients only found in infrastructure or advanced manufacturing. These uncertain environments make it difficult for developers to anticipate every possible program behavior, especially coupled with the intermittent execution of the device. Even with tools like Ekho [8], which record and replay these energy environments accurately, and simulators like SIREN [7], no tools can currently guarantee or explore all possibilities.

Timely Execution: Intermittent execution means that the execution time (from start to finish) of a specific task can vary widely. Long-running or expensive tasks can be interrupted by power failures of varying length, meaning that data or computation can expire before a task is completed. For example, a single acceleration reading may only be relevant for a few seconds, so a sensor needs to ensure that long running tasks using that data do not spend resources on it if the data is older than a few seconds. Hardware methods can provide a sense of time [10], but knowing how to manage that time, and expose the interface to the developer or user requires more thought. Timely execution affects everything from accuracy of sensor data, to user experience and network protocols.

Coordination and Sharing: Batteryless sensing will enable massive deployments of heterogeneous, networked sensors in environments with varied energy availability. The “networked” part of this statement requires more development. Networking traditional sensing devices with stable power and consistent execution is difficult. Batteryless sensors have no guarantees on communication or homogeneity of available resources. Therefore it is likely that a batteryless sensor broadcasting may have no one to hear it.

Usability and Experience: Developers and users of batteryless sensing devices must be able to program, test, and interact with these devices. Language models have been developed that try to simplify the memory model and execution flow of batteryless development [5]. Energy management techniques like Federated Energy [9] attempt to reduce failures and simplify programmer choices. However, the developer needs more than just language and hardware support, they need tools, platforms, and deployment ability that is both usable and intuitive. The user needs to be able to wear or interact with batteryless sensors despite the intermittent power supply, and expect that devices will work at some level of quality. These ideas need to be verified in user studies and experiences with real users and real applications.

Answering these challenges requires rethinking the sensing stack in three key new directions: (1) tools and the platforms they connect with, (2) languages and the runtime systems they rely on, and (3) networking protocols and the architectures they depend on.
3 PLATFORMS AND TOOLS

Standard platforms and usable tools enable developer communities to grow and thrive. Batteryless platforms are in their infancy, decreasing their utility, impact, and reach. The WISP [17] has been the platform of choice for research in batteryless devices, yet it only harvests from the RFID band. In this section we identify why building platforms and tools for batteryless computing is difficult, and outline research directions. We frame the discussions around the challenges and promise of batteryless sensing in the context of massive scale infrastructure and environmental monitoring. Figure 2 shows the broad application domain for these platforms and tools. Massive scale monitoring of the environment and infrastructure has been proposed as a way to extend the lifetime of national infrastructure and monitor the environment for pollutants and change. Two motivating examples are pipeline monitoring and air quality monitoring.

Both applications require a rethinking of sensor platforms and tools to enable the massive deployment of cheap, useful, maintenance-free sensors. These sensors must be batteryless because of the number of devices needed to make a worthwhile application. This means these sensors developers need to be able to carefully account for deployment conditions, so that only one deployment is needed, and maintenance is trivial. This means these developers need to be able to rapidly prototype and test multiple designs before deploying.

3.1 Standardized Prototyping Batteryless Sensing

One of the main obstructions keeping batteryless sensing from entering the mainstream of sensing is the lack of a standardized, flexible hardware platform. Batteryless sensing systems use a wide variety of energy sources and function in a very wide array of applications, often requiring specialized hardware (and software). Designing motes from scratch is costly, time consuming, and leaves little room for error. In the next five years, platforms that enable rapid prototyping, enable usage in multiple environments, and support a myriad of sensors must be developed for intermittent computing.

3.2 Accounting for Deployment Conditions

Batteryless sensors are meant to be maintenance-free because of the impossibility of physical access and time to access every device Therefore the tools and methods to ensure successful ongoing deployment are critical. Deploying vast amounts of sensors correctly—and verifying behaviors pre-deployment in the lab—must to be a focus of the next few years. Without a repeatable way to compare different algorithms, hardware platforms, and energy harvesters, batteryless sensing can have no rigorous scientific backing.

**Energy Environments:** Energy availability is a key determining factor in the behavior of a batteryless device. The behavior of a batteryless device has a measurable effect on the amount of energy that can be harvested. Behavioral changes of a sensor device—going from low power mode to broadcasting on the radio, sensing the temperature, or simple processing—cause the voltage on the harvester to change, which alters the harvesting efficiency. Ekho [8] records and replays these energy environments, while tools like the Energy-interference-free Debugger (EDB) provides GDB-like functionality for intermittently powered devices. Extensions of this level of pre-deployment testing and profiling is necessary to push batteryless sensing forward. Integrating these tools with energy aware forms of code coverage could provide developers with knowledge of how their devices will execute in varied deployment environments. Analytical models constructed from energy environments recorded by Ekho, could allow developers to speculate on the limits of energy availability, and then restructure or adapt their hardware designs and software before ever deploying. Managing and understanding energy environments is a crucial place for research in the next push for sophisticated batteryless sensors.

**Testbeds:** Continuing the tradition of the sensing community, we call for testbeds to support research in massive scale infrastructure monitoring with batteryless sensors. This is not just a minor extension of current testbeds—batteryless sensing testbeds must be able to emulate changing energy environments and the sensor data that changes with energy availability. Additionally, testbeds for Wireless Energy Transfer (WET) must be imagined to give agile and varied solutions to sensing problems. Testbeds must be energy source agnostic (allowing for emulation of solar, thermal, kinetic, and RF sources). Testbeds must allow per node inspection without interfering with the energy or data streams of sensors. Testbeds will allow for investigations in intermittent networking, measurement of ability in a large network, and explorations in redundancy.

3.3 Research Questions

We consider important research questions to spark discussion and guide others in search of new topics.

1. What information besides energy environments are needed to build robust analytical models of deployment conditions?
2. Can testbeds enable other areas like energy aware code coverage or massive scale scheduler testing?
3. How can we build a robust community—with tools and platforms—that engages people outside of sensing and embedded systems?
4. Can batteryless hardware platforms be developed that engage the user through displays and other mediums? What applications would this benefit?
Batteryless Sensing for the Built Environment

Figure 3: This figure shows the many applications for batteryless sensors in the built environment, specifically a home. These invisible, cheap sensors, can harvest indoor light, RF signals, vibrations or heat from appliances, or even energy from footsteps to power their tasks.

## 4 LANGUAGE, RUNTIME, OPERATING SYSTEM

Runtime operation through intermittent power failures is a cognitive burden on the developer as well as a resource drain on the device. Despite advances in tooling and hardware, developers still find programming batteryless devices very challenging and non-intuitive. Progress has been made—Chain is a task based language for intermittent computing that uses memory channels to simplify control flow and formalizes memory usage [5] to guarantee memory consistency. However, composing programs for batteryless, intermittently powered devices still requires in-depth knowledge of hardware and energy harvesting behaviors. Tackling the interfaces between the developer and the device are critical to enabling the smart home vision. Figure 3 shows many of the application domains in the built environment that tackling language, runtimes, and operating systems for batteryless sensors will enable.

Four main challenges in the context of language and runtime systems exist for anyone entering this new direction of batteryless research. We detail them—as well as potential research directions—below.

**Visualizing Control-Flow:** A power failure can happen between any two lines of code. This is a cognitive burden on developers who have little experience with intermittent computing. Much of the work involved in writing robust batteryless sensing code comes from guarding against failures. This means that application goals, defining tasks, and data generation, are all obscured. Even with Chain, programmers must be careful to divide up tasks in small enough portions so that tasks will finish. Developers must also take care to describe necessary data structures and channels in C, with control flow buried inside the task itself. Defining language models and tools that allow for visualizing and understanding the control flow of a program is a critical challenge to overcome so that new developers can quickly get started creating robust programs.

**Timing Failures:** Previous work in the batteryless language and runtime space has focused exclusively on preserving forward progress, assuming that continuing where you left off is a good thing. This is often a bad assumption—there are many cases where continuing old computation from a previous device lifetime is not good for application quality of service. Consider a sensor taking a mass of acceleration readings in an activity detection application. If the device loses power shortly after gathering the readings, and is unfortunate enough to not resume computation until an hour later, why would it waste compute and energy on processing those readings for an activity that happened an hour ago? The challenge of integrating time as a way to conserve resources and good throughput remains open.

**Distributed Resources:** Previous work has ignored the fact that sensing devices are not just a computational unit—they are made up of potentially many devices, the microcontroller (MCU), energy harvester, power supply, multiple sensing units (for instance a magnetometer, accelerometer, or temperature sensor), external storage, even actuators. Each of these devices have specific energy requirements (tied to the task the developer wants to execute), specific operating requirements, and sometime their own volatile or non-volatile memory. *Every sensor is a distributed system*—the sensors is not just a microcontroller. The reason for treating batteryless sensors as just a microcontroller is because the first platforms (like the WISP[17]) were RFID powered, and only had an ultra low power accelerometer and a MCU that were closely tuned in terms of operating range and energy requirements, meaning that it was simple to avoid the distributed nature. As batteryless devices tackle more applications and integrate more complex components this will not always be true. Runtime techniques that capture this distributed nature and make it simpler to schedule tasks are critical going forward.

### 4.1 Research Questions

Taking these challenges together, we consider important research questions and directions to spark discussion and guide others in search of new topics.

1. Batteryless devices are intended to be deployed for decades—therefore reprogramming will be essential to fight off obsolescence. Are their programming models or runtime techniques that make reprogramming easier in an intermittent and energy starved environment?
2. Information for batteryless sensors that enable better scheduling is very local, but operation is mostly determined by global factors like environmental conditions. Can sharing or macro-languages bridge this divide in information?
3. Can approximate computing techniques be applied to meter energy efficiency, guaranteeing some quality of application in the face of changing energy environments? What are the indicators that tell programs to change? What is the programming model that encapsulates user definitions of quality of service? What tasks or sensors can be downgraded or upgraded in service quality?
4. How should energy be intuited in the programming model? Should it be exposed at all to the programmer? Is it a distraction?
5 NETWORKING PROTOCOLS AND ARCHITECTURES

Batteryless sensor networks do not currently exist because of the incredible difficulties of coordinating and syncing multiple intermittently powered ultra constrained devices. Batteryless nodes in a network do not have an accurate sense of time, have limited computational abilities, and do not have the energy storage to support radio activity for long lengths of time or over large distances. Despite these challenges, these networks would be of significant value to the Internet-of-Things—therefore, overcoming these challenges is crucial.

Batteryless sensor networks enable broad applications across sensing, in this section we focus on enabling the body area network (shown in Figure 4). Wearable body networks enable monitoring for users over longer periods of time than a smartphone or battery powered wearable (like a smartwatch) can provide. These devices can be placed invisibly, and in large numbers. These sensors can harvest energy from light sources, thermal variations, in-place RFID readers, or the movement of the worker. Behavioral psychologists and medical practitioners can automate Ecological Momentary Assessments (EMAs) over many months, allowing for longterm assessment of behavior. Numerous invisible sensors in clothing, jewelry, or the workplace can coordinate to deliver health and fitness metrics to the provider and patient. In each of these applications batteryless sensors allow denser, longer, and less obtrusive deployments in applications that have been impractical with traditional sensing.

In the rest of this section we focus on enabling the body area network. These tiny devices unobtrusively monitor the health and wellness of a worker or patient, and can provide interventions from medical care providers, managers, or automatically—while assisting and interacting with the wearer from a smartwatch hub.

5.1 Intermittent Network Protocol

The constrained resources and intermittent nature of batteryless sensors requires a rethinking of sensor network assumptions around idle listening, synchronization, overhearing, collisions, and even network topology. Idle listening with active radios—a key part of most protocols for traditional battery powered sensors—is not feasible with the limited energy budgets of batteryless sensors. Synchronization becomes difficult because losing power for indeterminate lengths means heartbeat style transmission are often impossible to maintain (since energy will not always be available). This can make common MAC layer protocols like TDMA difficult to implement without modification. Additionally, network topology can change while a sensor is in a power failure—so local information about network topology cannot be trusted immediately after a power failure, even with accurate timing information available locally. Collisions become problem as these networks grow and become more dense. Many sensors all fight for limited space and with limited resources and information for coordination.

Batteryless deployments offer unique attributes that can be leveraged for robust protocol. Energy environments are often gradient across a geographic area, so bands exist of different energy availability. This is most evident with solar or RFID energy sources. As an RFID reader moves across a network of sensors, those closest to the reader will have the most energy, this means that collisions and overhearing of transmissions in tight geographically located areas are likely. If a sensor is able to broadcast, then it’s nearest neighbors are likely able to as well. This assumption that sensors in the same neighborhood have similar energy could be used for coordination.

Protocols for intermittent devices must be aware of the complications from the intermittent computing model. New protocols must take advantage of the things that batteryless networks have to offer, namely redundancy, denseness of deployment, and the link between energy availability and location. Protocols will need to move away from heartbeat models and adopt disruption tolerant models.

5.2 Architectural Support

Architectural and hardware support for batteryless sensors can simplify the difficulties outlined in defining and enforcing protocols in batteryless, intermittent, sensor networks. Key advances in backscatter [12] and passive Wifi [18] will enable devices to communicate without active radios as long as a modem or basestation is present. However, this model will not work for all deployment scenarios, especially outside of built environments. Passive listening techniques could be promising for enabling a form of idle listening currently used in battery powered sensors. Pursuing passive listening techniques, energy aware transmissions, and developing hardware platforms that provide hardware interrupts for radio use is a critical area of exploration in the next few years for batteryless networked sensor architecture.

5.3 Research Questions

Taking these challenges together, we consider important research questions and directions to spark discussion and guide others in search of new topics.
(1) Synchronization is difficult without accurate, always on clocks, can energy events that affect large numbers of co-located sensors be used as synchronization events? For instance the setting or rising of the sun, or the passing of the RFID reader?

(2) As the number of batteryless devices start to exceed a trillion, how does the scale affect the protocol and architecture of the individual sensor, the gateways, and the user or developer?

(3) In a large network of batteryless sensors, there are bound to be many different types of resources available, but spread out across nodes. How can sensors coordinate and share hardware and compute resources across an intermittent network for the betterment of the wearable user, the infrastructure, or the building manager?

(4) Intermittently powered body area health networks transfer private data across many intermittent nodes, how can we ensure this data is protected and secured based on user input while not constraining operation?

6 CONCLUSIONS

This paper has outlined three new directions for research in support of batteryless, intermittent sensing. Batteryless sensors promise to bring about the vision of “smart dust” proposed decades ago, where trillions of invisible devices interact with each other, with people, with the environment, and with infrastructure to benefit our daily lives and society. These sensor harvest energy, compute opportunistically, and fail intermittently when no energy is available. This intermittent computing model complicates everything from design, to debugging, to deployment. However, these devices enable exciting new applications in many untapped domains—including truly perpetual infrastructure monitoring, body area health networks, large scale sensing deployments, and even space exploration and sensing in extreme environments. We believe innovation in (1) standardized hardware platforms and tools, (2) languages and runtime systems, and (3) networking protocols and architectures are key to the success of sensing in the next five years. We call the community to action—developing and then sharing with the community open-source and open-hardware research artifacts in support of batteryless computing. We believe we can advance this vision of sensing and catapult the embedded networking sensing community into the next bright future of computing.

ACKNOWLEDGMENTS

This research is based upon work supported by the National Science Foundation under grant CNS-1453607. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

REFERENCES


