Continuous Sensing on Intermittent Power

Amjad Yousef Majid
a.y.majid@tudelft.nl
Delft University of Technology

Patrick Schilder
p.t.schilder@student.tudelft.nl
Delft University of Technology

Koen Langendoen
k.g.langendoen@tudelft.nl
Delft University of Technology

Stephan Wong
j.s.s.m.wong@tudelft.nl
Delft University of Technology

ABSTRACT

The vision of ubiquitous sensing runs into the reality of battery technology. Batteries are short-lived (about 14,000 tons of rechargeable batteries are thrown away in United States alone); hazard; and costly—costs include manufacturing, replacement, and disposing. Batteryless sensors power themselves from ambient energy. Ambient energy is marginal and unpredictable. Consequently, tiny batteryless sensors operate intermittently, when sufficient energy, to do a simple task, is buffered, execution begins and terminates shortly after to recharge the depleted buffer. Sporadic sensing does not meet a wide range of real-world applications; therefore, intermittent sensors have not been widely adapted. We present *Coalesced Intermittent Sensor* (CIS) an intermittent sensor that sense continuously! The CIS design takes advantage of the randomization embedded into the powering subsystem—an energy source and an energy harvester—to distribute its nodes uptime. Additionally, we realize the CIS in a prototype of a batteryless voice assistant agent. Our results show that x% of the commands are captured [uptime results].

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Embedded systems, Distributed intermittent computing

ACM Reference Format:
Amjad Yousef Majid, Patrick Schilder, Koen Langendoen, and Stephan Wong. 2019. Continuous Sensing on Intermittent Power. In *Proceedings of ACM Conference (Conference'19).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

[Make it clear that the goal of this work is continuous sensing] [Explain that we approach continuous sensing on two stages: 1) continuous availability and 2) randomized response]

Conference'19, July 2019, New York, NY, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

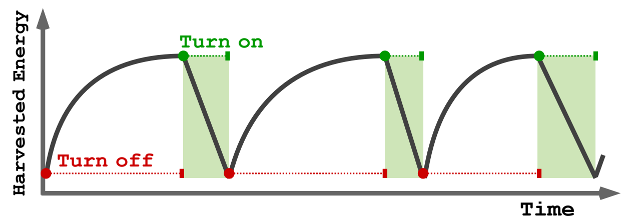

Figure 1: The power cycle of an intermittently powered device.

The Internet of Things is the engine that will drive smart cities. In these cities, cars will not need to wait in front of traffic lights for non-existing pedestrians to cross the road; doors, upon leaving, will provide people with weather forecast; and jackets will adjust air circulation based on body temperature. Smart cities will gain their awareness through billions of sensors.

Battery-powered sensors do not provide a viable solution to power all the sensors in smart cities. Batteries require (i) regular maintenance, even rechargeable ones wear out in a few years [**?**]; and (ii) hazard waste management. Moreover, batteries raw materials are also limited. Therefore, the edges of the future Internet of Things must leave batteries behind and rely on perpetual energy sources.

Natural energy sources such as light, vibration, and heat can power tiny sensors directly. Tiny energy harvesters, however, can only scavenge a very limited power from such energy sources. Therefore, an energy-harvesting sensor operate intermittently. An intermittent sensor starts by harvesting a certain amount of energy, in its buffer (i.e. a super-capacitor). Then, it triggers operation which depletes the buffered energy quickly, as the power consumption rate tends to be much higher than the power accumulation rate. Once the energy is below a certain level, the sensor experience a complete power-down, the cycle of charging and operating continues indefinitely (Figure 1).

Intermittent nodes trade-off reliability (reliable energy source) for sustainability (sustainable energy source for a very large number of sensory nodes). This trade-off generates many challenges. For example, preserving forward computation progress, enabling timely operations, and the fact that nodes are not always available.

Many of these challenges have been tackled. For example, the intermittent computation problem, which is concerned with the preservation of an application progress and memory integrity under frequent power failures [4, 20, 21]; timely operation, which is concerned with data freshness after a power interrupt [8]; and event-driven execution, which is concerned with input output operations under arbitrarily-timed power loss [?].

Despite the significant progress that has been achieved in intermittent domain, *the system availability problem* has not been addressed. A monitoring sensor that has a very low probability to be available when an external event occurs is not worth deploying. A sensor that is capable of capturing only very short events has a limited number of potential applications (imagine that you want to control a room lights with batteryless microphone. The microphone is capable of processing a single word. If you say "on" lights turns on but other systems might start operate also, a specification problem. If you say "light" to eliminate other systems you lose the control functionality). Consequently, intermittent sensors have not gained widespread adaptation.

This paper tackles the paradox of continuous sensing on intermittent devices by introducing a new type of sensors that we call *Coalesced Intermittent Sensor* (CIS). The CIS is defined as a group of intermittent nodes with randomized on/off cycles. CIS distinguishes between different energy conditions and adapts its response accordingly to efficiently distribute its resources and to approach continuous sensing.

We put our observations and theory into test by realizing a CIS instant in the form of distributed intermittent voice assistant agent. We tested the voice assistant in different energy conditions and the results validate our assumptions and observations.

Highlights of the paper contributions:

- Introducing a new sensor type: *Coalesced Intermittent Sensor* (CIS) is a group of intermittently powered nodes that takes advantage of the randomized nature of the powering subsystem (energy source and energy harvester) to approach 100% system availability. CIS has a sensing algorithm that is energy harvesting conditions aware and therefore it is an efficient sensor.
- Characterizing the behavior of coalesced intermittent sensor under different energy conditions.
- Characterizing the behavior of coalesced intermittent sensor under different event occurrence frequencies and patterns.
- Implement a coalesced intermittent sensor in the form of a distributed intermittent microphone array.

[to be removed!] It increases the temporal and spatial availability of an intermittent system and enables resource distribution such as large number of words templates for spoken words recognition systems.

Controlling the on/off cycle of intermittent devices enables adapting them to many real world applications. For example, once a certain on/off cycle is preserved, an intermittent wake-up receiver can be implemented; intermittent acoustic monitoring system for monitoring engines modules—the sound produced by a deformed gear tooth—can be made. Moreover, with the advances in passive communication (such as passive light [], and backscatter tag-to-tag [] communication) battery-free miniaturized sensors can form self-powered wireless sensor network to, for instance, create smart wallpaper and revolutionize smart buildings.

2 COALESCED INTERMITTENT SENSING

Coalesced Intermittent Sensor(CIS) is the abstraction of a group of batteryless intermittent sensors. CIS orchestrates its nodes power cycles using a distributed approach (instead of relying on a master powerful node to coordinate coalesced nodes activities).

2.1 Intermittent Nodes' Power Cycles Distribution

To ensure an efficient distribution of the intermittent nodes' uptimes, an explicit or implicit control methods can be applied: (i) explicit control of nodes' power cycles requires inter-nodes communication. Communication between batteryless nodes requires ultra low power communication regimes to be efficient. Recent advancements in passive visible light communication [?] and ambient radio frequency backscattering [] demonstrate the feasibility of extremely energy efficient communication between batteryless nodes. Once messages exchange is possible, an intermittent node duty is to estimate the number of active nodes in its time slot and to decide on leaving this power cycle of maintaining it (Figure ??). Nodes can influence their power cycles by altering their load—low power consumption extends a node's uptime and delays its off-time, and high power consumption has an inverse effect. (ii) implicit control of nodes' power cycles approach seeks to use a random process to "ideally" uniformly distribute nodes' on/off cycles over the CIS's power cycle: when all the nodes complete their individual power cycles. CIS takes advantage of the randomized nature of the ambient energy to distribute its coalesced nodes.

In this paper we opt to explore the implicit nodes distribution control approach as it is simpler and requires less overhead than explicit control methods [1].

Implicit Control: Exploiting Solar Power. The implicit control methods have the advantage of not requiring control messages exchanging and processing.

Figure 2 shows an implicit solar power based CIS's nodes distribution when they are powered by artificial light (during night) and sunlight (during day).

2.2 Sensing on Intermittent Power

3 BATTERYLESS DISTRIBUTED MICROPHONE

CIS is a prototype of a distributed battery-free audio assistant agent. CIS is envisioned to be embedded in, for example, wallpapers and furniture coverage to turn these objects to smart and controllable ones. The distributed nature of the CIS system enables it to exploit the inherent randomness of intermittent devices to offer a reliable interaction.

CIS is a battery-less distributed microphone. It operates intermittently, when a threshold amount of energy accumulates.

Audio is recorded for a fixed time period. After that, if the audio recording has finished successfully, the recording is divided into smaller parts called *frames* and speech recognition is performed

[1]The hardware used to demonstrate the feasibility of passive light communication and ambient RF backscattering are not open source and re-making it is beyond the scope of this study [add the references to Marco's paper and ambient RF paper]

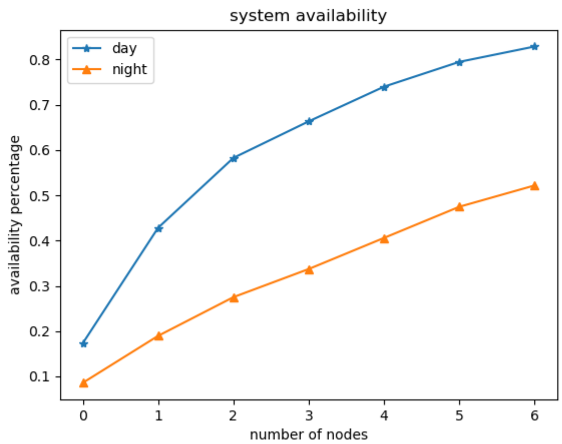

Figure 2: Implicit Coalesced Intermittent Sensor' nodes distribution using solar power harvesting randomization

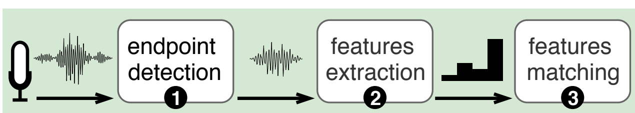

Figure 3: (1) endpoint detection algorithms extract the part of the signal corresponds to the input word, (2) feature extraction calculates the energy spectrum of the trimmed signal, and (3) feature matching searches the local database for the closest word

using an algorithm consisting of three steps: (1) endpoint detection (2) feature extraction (3) feature matching. This steps are also depicted in figure 3.

During *feature extraction* on frames containing the word Fast Fourier Transform is applied to obtain the energy spectrum. The energy spectrum is then sorted into a feature vector, where each vector value holds the total energy of a specific frequency range. Finally each vector value is normalized over the total vector energy.

After that, *feature matching* is performed, where the features of the recorded word are compared against a local database to find the most similar word in the database. There exist many feature matching methods, yet only few are able to run under the constraints of an intermittent system. Two methods have been implemented and tested. The first is Dynamic Time Warping (DTW), which is able to compare two sequences of data, even when they vary in speed. This way if a word is spoken slower or faster, the algorithm can compensate for that. The second is a method that linearly compares two feature vectors. This requires less computations than DTW.

3.1 Hardware description

we used MSP430RF5994 [27] an ultra-low-power microcontroller to drive the PMM-3738-VM1010-R microphone that features zero power listening and wake-on-sound technology [19]. A BQ25570EVM-206 [26] solar power harvester connected to a pair of SLMD121H04L [10] solar panels intermittently powers a CIS sensor. For some experiments simulated intermittent power is used. For debugging, we used saleae logic analyzer [23].

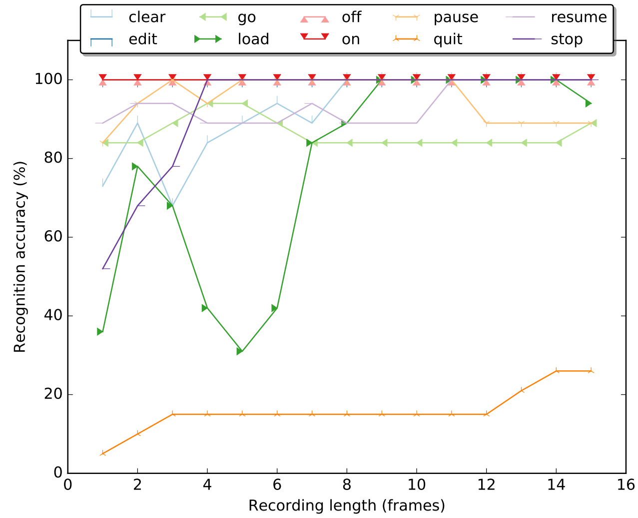

Figure 4: Recognition accuracy versus the recording length (in frames), while using multiple recordings per word for testing. Used feature matching method: linear.

3.2 Implementation

During recording a sample rate was used of 7812 Hz, which covers the human voice frequency range. The used frame size was 256 samples. This size was beneficial for doing a Fast Fourier Transform and corresponds to approximately 33 milliseconds of speech.

For normalization an integer log function was applied on every value, divided by the average log. The whole algorithm is implemented using only integer / fixed point arithmetic.

In the linear comparing method the feature vectors of two words are compared successively, not accounting for differences in the speed of pronunciation. If two words vary in length, the last frames cannot always be compared and instead a penalty is applied linear to the length difference.

In between the different steps (see figure 3) checkpoints in non-volatile memory are used to assure progress while running on intermittent power. In some cases additional checkpoint were used inside the steps.

4 EVALUATION

Table 1: Testing set

on	off	stop	clear	load
go	pause	resume	edit	quit

4.1 Effective recording length

Since our target hardware has extremely limited resources, the first experiment targets the minimum effective recoding length without significant accuracy loss.

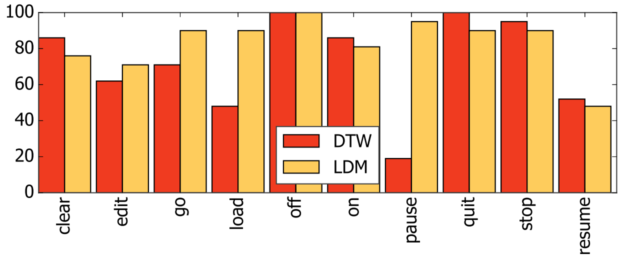

Figure 5: Recognition accuracy for linear matching and DTW when using 9 frames as the recording length.

Table 2: Profiling of features matching algorithms: Dynamic Time Warping (DTW) and Linear Distance Matching (LDM)

Section	Linear (ms)	DTW (ms)
Recording	285.9	285.9
Feature extraction	501.9	501.9
Feature matching	99.4	1251
Total	887.2	

Table 3: The mean and standard deviation of the parameters that were used to simulate intermittent execution. Here t_{on} is the time the device is on while recording or processing, t_{off} is the time the device is charging and t_{sleep} is the time the device is sleeping while waiting on sound input.

Parameter	μ (ms)	σ (ms)
t_{on}	590	17.7
t_{off}	5310	154
t_{sleep}	22420	652

For this experiment a single microcontroller running on continuous power is used. Each word from Table 1 was recorded on a PC 20 times. The features—normalized FFT-based values—of the first recording are saved in the microcontroller's persistent memory as a signature to perform the feature matching, during the testing. the rest of the recording are used for conducting the experiment.

Figure 4 shows words recognition accuracy when the 19 recordings of each word are played back from the PC speaker. We can concluded that recording beyond *nine frames* do not increase the recognition accuracy; therefore, nine frames recording length is chosen for the rest of the experiments.

4.2 Comparison of feature matching methods

We implemented two algorithms for voice features matching, Dynamic Time Warping (DTW) and Linear Distance Matching (LDM). Due to the microphone wake-on-sound feature and the fixed recording length, DTW did not outperform LDM as Figure 5 shows. Moreover, DTW takes more time to process that data as Table 2 shows. Therefore, LDM algorithm is used in future experiments.

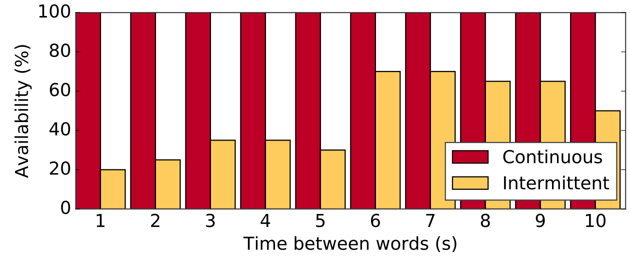

Figure 6: The effect of the time between consecutive words on the availability: the percentage of words that is processed by the command recognizer. [Add correct recognition as stacked bar?]

4.3 Intermittent microphone availability

This experiment shows the effect of intermittent power supply on a microphone availability [and recognition accuracy] for different command (or word) repetition speed. Each word (from Table 1) was played back 20 times with intra-words-playing time ranges from 1 to 10 seconds.

For the experiment we chose to simulate the intermittent power supply, based on a real measurement, to ensure environment consistency across different words. In particular, we measured the t_{on}; used an on/off ratio of 10% to simulate harsh harvesting conditions [21? ?]; and calculated the t_{sleep} based on the micro controller specification [?] (Table 3).

The obvious observation from Figure 6 is that intermittency has a great impact on the command recognizer availability. However, a less obvious, but important, observation is that the correlation between the on/off cycle and the command repetition speed has also great impact as the bars labeled with 5 and 6 from Figure 6 indicate.

5 BACKGROUND

5.1 Energy harvesting

Ambient energy is volatile, and scarce. For example, radio waves harvestable power varies from nW-scale when harvesting ambient RF energy to µW-scale when harvesting a dedicated RF signal; and solar power ranges from tens of µW to tens of mW when it is harvested by a solar panels of a few cm^2 illumination surface [13, 22].

A tiny energy-harvesting device slowly charges its energy buffer (e.g. capacitor) while the device is off. Once the buffer is full, the device begins operating and depleting its energy reservoir—since power consumption is much higher than harvested power—until it powers down. This charging-discharging cycle repeats indefinitely (Figure 1).

5.2 Intermittent computing

Intermittent execution models [2, 4, 14, 28] enable applications to progress despite frequent power failure. They decompose an application into several small code pieces and save the progress state of the application on the transitions between these code segments. Therefore, intermittent applications do not return to the same execution point (e.g. `main()`) after each power failure, as the applications that assume continuous power supply, instead they resume from the last saved progress state of execution.

5.3 Spoken words recognition

Speech recognition algorithms can be classified based on the type of speech that they can recognize into: *spontaneous speech, continuous speech, connected word,* and *isolated word* [7].

Systems with *continuous* or *spontaneous speech* recognition are the closest to natural speech, but are the most difficult to create because they need special methods to detect word boundaries [7]. This is less the case for the *connected word* type, where a minimum pause between the words is required. The type with the least complexity is the *isolated word* type. It requires a period of silence on both sides of a spoken word and accepts only single words.

Speech recognition consists of several steps. The basic steps are mentioned briefly here: *Speech recording and signal digitization*—a microphone records the sound waves and an ADC converts the microphone signal into a digital signal. A sampling rate of about 8 kHz is required to capture the frequencies of a human voice (100-4000Hz [**?**]). *Framing*—after that the digitized signal is divided into blocks of usually 10-30 ms [5–7] called frames. *Features extraction*—for each frame a feature vector is extracted containing all the relevant acoustic information. *Feature matching*—finally the extracted features are matched against features known to the recognizer.

6 RELATED WORK

6.1 Energy-harvesting

Many batteryless energy-harvesting platforms have been proposed, for example Wireless Identification and Sensing Platform (WISP) [24] and its variants such as NFC-WISP [34], WISPcam [16], and NeuralWISP [30], batteryless phone [25], ambient backscatter tag [12] and Moo [32].

6.2 Intermittent execution

[brought from introduction]Mementos [21]proposed a Checkpointing-based approach to enable long-running applications on intermittently-powered devices. DINO [20]enables safe non-volatile memory access despite power failures. Chain [4] minimizes the amount of data need to be protected by introducing the concepts of atomic tasks and data-channels. Alpaca and Ratchet [15, 29] use compilers to automate intermittent code generation. Mayfly [**?**] enables time-aware intermittent computing. InK [31] introduces event-driven intermittent execution.

Intermittent systems are regarded as the successor of energy-aware systems. Dewdrop [3] is an energy-aware runtime for (Computational) RFIDs such as WISP. Dewdrop goes into low-power mode until sufficient energy for a given task is accumulated. QuarkOS [33] divides the given task (i.e. sending a message) into small segments and sleeps after finishing a segment for charging energy. However, these systems are not disruption tolerance.

The first power-failure-tolerant systems use the idea of volatile progress state checkpointing into persistent memory [21]. DINO [20], however, shows that in addition to the volatile memory, the non-volatile memory of the processor must also be protected to ensure correct executions. Hibernus [1] measures the voltage level in the energy buffer to reduce the number of checkpoints. Ratchet [29] uses compiler analysis to eliminate the need of programmer intervention or hardware support. HarvOS [2] uses both compiler and hardware support to optimize checkpoint placement and energy consumption.

Task-based systems optimize intermittent execution by reducing the amount of data needed to be saved into non-volatile memory to protect applications against power interruptions [4]. However,

6.3 Speech recognition

The speech recognition problem has been tackled from many angles and has experienced many great breakthroughs. For example, Dynamic time warping (DTW) algorithm enables matching voice signals with different speed (or time) []. Approaches based on Hidden Markov Models showed much better performance than DTW-based ones [11]. Hence, they became the standard techniques for general purpose speech recognition until artificial intelligent algorithms [9], however, outperform them.

Many specialized hardware architectures for speech recognition have been proposed to, for instance, reduce energy consumption [17, 18].

7 DISCUSSION AND FUTURE WORK

8 CONCLUSION

REFERENCES

[1] Domenico Balsamo, Alex S. Weddell, Geoff V. Merrett, Bashir M. Al-Hashimi, Davide Brunelli, and Luca Benini. 2015. Hibernus: Sustaining Computation During Intermittent Supply for Energy-harvesting Systems. *IEEE Embedded Syst. Lett.* 7, 1 (March 2015), 15–18.

[2] Naveed Anwar Bhatti and Luca Mottola. 2017. HarvOS: Efficient code instrumentation for transiently-powered embedded sensing. In *Information Processing in Sensor Networks (IPSN), 2017 16th ACM/IEEE International Conference on*. IEEE, 209–220.

[3] Michael Buettner, Ben Greenstein, and David Wetherall. 2011. Dewdrop: an Energy-aware Runtime for Computational RFID. In *Proc. NSDI*. USENIX, Boston, MA, USA, 197–210.

[4] Alexei Colin and Brandon Lucia. 2016. Chain: tasks and channels for reliable intermittent programs. *ACM SIGPLAN Notices* 51, 10 (2016), 514–530.

[5] Brian Delaney, Nikil Jayant, Mat Hans, Tajana Simunic, and Andrea Acquaviva. 2002. A low-power, fixed-point, front-end feature extraction for a distributed speech recognition system. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, Vol. 1. IEEE, I–-793.

[6] Brian Delaney, N Jayant, and T Simunic. 2005. Energy-aware distributed speech recognition for wireless mobile devices. *IEEE design & test of computers* 22, 1 (2005), 39–49.

[7] Santosh K Gaikwad, Bharti W Gawali, and Pravin Yannawar. 2010. A review on speech recognition technique. *International Journal of Computer Applications* 10, 3 (2010), 16–24.

[8] Josiah Hester, Kevin Storer, and Jacob Sorber. 2017. Timely Execution on Intermittently Powered Batteryless Sensors. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 17.

[9] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29, 6 (2012), 82–97.

[10] IXYS Corporation. 2018. IXOLAR™ High Efficiency SLMD121H04L Solar Module. http://ixapps.ixys.com/DataSheet/SLMD121H04L_Nov16.pdf

[11] Frederick Jelinek. 1997. *Statistical methods for speech recognition*. MIT press.

[12] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. 2013. Ambient backscatter: wireless communication out of thin air. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 39–50.

[13] Brandon Lucia, Vignesh Balaji, Alexei Colin, Kiwan Maeng, and Emily Ruppel. 2017. Intermittent Computing: Challenges and Opportunities. In *LIPIcs-Leibniz International Proceedings in Informatics*, Vol. 71. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[14] Brandon Lucia and Benjamin Ransford. 2015. A simpler, safer programming and execution model for intermittent systems. *ACM SIGPLAN Notices* 50, 6 (2015), 575–585.

[15] Kiwan Maeng, Alexei Colin, and Brandon Lucia. 2017. Alpaca: intermittent execution without checkpoints. *Proceedings of the ACM on Programming Languages* 1, OOPSLA (2017), 96.

[16] Saman Naderiparizi, Aaron N. Parks, Zerina Kapetanovic, Benjamin Ransford, and Joshua R. Smith. 2015. WISPCam: A Battery-Free RFID Camera. In *Proc. RFID*. IEEE, San Diego, CA, USA, 166–173.

[17] Michael Price, James Glass, and Anantha P Chandrakasan. 2018. A Low-Power Speech Recognizer and Voice Activity Detector Using Deep Neural Networks. *IEEE Journal of Solid-State Circuits* 53, 1 (2018), 66–75.

[18] Michael Price, James R Glass, and Anantha P Chandrakasan. 2015. A 6 mW, 5, 000-Word Real-Time Speech Recognizer Using WFST Models. *J. Solid-State Circuits* 50, 1 (2015), 102–112.

[19] pui audio, vesper. 2019. PMM-3738-VM1010-R: A ZeroPower Listening piezoelectric MEMS microphone. http://www.puiaudio.com/pdf/PMM-3738-VM1010-R.pdf

[20] Benjamin Ransford and Brandon Lucia. 2014. Nonvolatile memory is a broken time machine. In *Proc. MSPC*. ACM, Edinburgh, United Kingdom.

[21] Benjamin Ransford, Jacob Sorber, and Kevin Fu. 2012. Mementos: System support for long-running computation on RFID-scale devices. In *Proc. ASPLOS*. ACM, Newport Beach, CA, USA, 159–170.

[22] VS Rao. 2017. Ambient-Energy Powered Multi-Hop Internet of Things. (2017).

[23] Saleae. 2017. Logic 16 Analyzer. http://www.saleae.com. Last accessed: Jul. 28, 2017.

[24] Joshua R. Smith, Alanson P. Sample, Pauline S. Powledge, Sumit Roy, and Alexander Mamishev. 2006. A Wirelessly-Powered Platform for Sensing and Computation. In *Proc. UbiComp*. ACM, Orange County, CA, USA.

[25] Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua R Smith. 2017. Battery-free cellphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 25.

[26] Texas Instruments. 2018. Ultra Low Power Management IC, Boost Charger Nanopowered Buck Converter Evaluation Module. http://www.ti.com/tool/BQ25570EVM-206

[27] Texas Instruments. 2019. MSP430FR5994 16 MHz Ultra-Low-Power Microcontroller Product Page. http://www.ti.com/product/MSP430FR5994

[28] Joel Van Der Woude and Matthew Hicks. 2016. Intermittent Computation without Hardware Support or Programmer Intervention.. In *OSDI*. 17–32.

[29] Joel Van Der Woude and Matthew Hicks. 2016. Intermittent Computation Without Hardware Support or Programmer Intervention. In *Proc. OSDI*. ACM, 17–32.

[30] Daniel J Yeager, Jeremy Holleman, Richa Prasad, Joshua R Smith, and Brian P Otis. 2009. Neuralwisp: A wirelessly powered neural interface with 1-m range. *IEEE Transactions on Biomedical Circuits and Systems* 3, 6 (2009), 379–387.

[31] Kasım Sinan Yıldırım, Amjad Yousef Majid, Dimitris Patoukas, Koen Schaper, Przemyslaw Pawelczak, and Josiah Hester. 2018. Ink: Reactive kernel for tiny batteryless sensors. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 41–53.

[32] Hong Zhang, Jeremy Gummeson, Benjamin Ransford, and Kevin Fu. 2011. *Moo: A Batteryless Computational RFID and Sensing Platform*. Technical Report UM-CS-2011-020. UMass Amherst.

[33] Pengyu Zhang, Deepak Ganesan, and Boyan Lu. 2013. Quarkos: Pushing the operating limits of micro-powered sensors. In *Proceedings of the 14th USENIX conference on Hot Topics in Operating Systems*. USENIX Association, 7–7.

[34] Yi Zhao, Joshua R Smith, and Alanson Sample. 2015. NFC-WISP: A sensing and computationally enhanced near-field RFID platform. In *RFID (RFID), 2015 IEEE International Conference on*. IEEE, 174–181.