Cooperative Sensor Networks for Long-Term Biodiversity Monitoring

Josh Millar, Sarab Sethi, Hamed Haddadi
Imperial College London

Michael Dales, Anil Madhavapeddy
University of Cambridge

We introduce *Terracorder*, a prototype multi-sensor device that uses collaborative on-device scheduling to improve its resource usage and extend its lifetime. The Kunming-Montreal Global Biodiversity Framework sets ambitious targets for 2030, including entirely halting human-induced species extinctions. Achieving these requires comprehensive data on global biodiversity patterns, which can only be obtained through in-situ distributed sensor networks. However, these multi-device networks are constrained by battery lifetimes, must gather rich data from power-hungry sensors, and yet need to be deployed in remote environments for long periods. Terracorder offers a low-carbon approach to long-term biodiversity monitoring, facilitating comprehensive data collection at minimal environmental impact.

1 Introduction

The low-power operation of battery-powered or energy-harvesting sensors is vital for long-term networked deployments. This is especially important for biodiversity monitoring networks, which are often deployed in remote, infrastructure-scarce environments, yet must gather rich data from power-hungry sensors (i.e., cameras and microphones). Consequently, biodiversity monitoring devices face a trade-off between data quality and coverage [18, 14, 2]; single-sensor devices [6] are cost-effective and low-power but offer limited data richness, while multi-sensor devices provide richer data but are generally expensive and power-hungry. We introduce *Terracorder*, a versatile multi-sensor device supporting low-power operation via adaptive scheduling. The application of biodiversity monitoring provides a unique opportunity here; learning schedules for events of interest (animal vocalizations, feeding activity, etc.) can not only help with optimizing power usage, but also generate large-scale information useful for conservationists.

Various strategies for adaptive scheduling include context rule-based methods [13], genetic algorithms [5], probabilistic approaches for estimating event likelihoods [8], and reinforcement learning-based schedulers [7]. The latter excel in the absence of predefined event models, especially if deploying many sensors for long durations, as event patterns vary and performance can degrade if scheduling isn't regularly fine-tuned. However, learning-based approaches for network-wide scheduling generally need federated/centralized coordination [11].

We instead look at on-device scheduling for event-driven networks, using low-power collaboration between neighboring devices to minimize redundancy. We detail a functional Terracorder prototype, including camera, microphone, and PIR, and evaluate its operational lifetime using real-world biodiversity data, power measurements, and a learning-based scheduler. Whilst initially isolated/standalone, its implementation lays the groundwork for a large-scale networked approach.

2 Prototype and Results

Prototype Design. Our Terracorder prototype is built on an ESP32s3-variant[1], and features ultra-low deep-sleep consumption of ~19μA (with 3.7V supply). This is much lower than other development boards with similar capacity, including other ESP32 variants.

The prototype also includes built-in power management features, such as a LiPo/LiIon battery charger IC and fuel gauge, supporting battery health monitoring and time-to-empty/full battery estimation along with a DC input to source reliable renewable power. The ESP32s3 itself is based on a XTensa SoC, supporting 512KB of internal SRAM and 16KB RTC SRAM for retaining state over deep-sleep. The SoC includes a RISC-V/FSM ultra-low-power co-processor for interfacing with external I2C sensors while its main processor remains in deep-sleep.

The prototype also includes a 5MP camera[2], an omnidirectional microphone[3], and PIR[4] sensor. These components were selected for their low-power operation and use in related biodiversity monitoring applications [3]. The PIR sensor remains active continuously, even in deep-sleep, acting as an event-trigger for our camera. The microphone is our scheduled component, but recording could also be event-triggered if events are of longer duration.

WiFi is used for transmitting recordings and images.

We measure the current draw of the various device modes using a high-voltage power monitor capable of μA-scale measurements[5]. The board is supplied at 3.3V via its JST-PH battery connector.

Table 1: Current measurements (3.3V in)

Mode/Operation	Current Draw (mA)
Deep-sleep (PIR and RTC active)	0.097
Microphone	
- 3s recording	31.57
- 0.1s recording + Goertzel filter	32.34
- 0.1s recording + TFLite inference	33.11
Camera (one activation)	49.33
Transmission (via WiFi)	
- 3s audio recording	61.33
- 5MP image	97.73
QL inference	0.031
QL update	0.071

Event Detection and Scheduling. We apply an off-the-shelf bird vocalization detection model, BirdNET [9], to generate events and durations from continuous audio recorded in the Sabah rainforest, Borneo[6]. We address BirdNET false positives by applying a

[1]https://powerfeather.dev
[2]Omnivision OV5640
[3]INMP441
[4]PaPIR EKMB1103111
[5]https://www.msoon.com/high-voltage-power-monitor
[6]The SAFE project

confidence threshold of 0.7. Users can generate species-specific thresholds based on expert validation of recorded pilot data to further improve detection accuracy [16]. The generated events are used in building and evaluating our scheduler. These, combined with our power measurements above, allow us to estimate our prototype's battery lifetime in a real-world deployment.

Our scheduler utilizes Q-learning [15], a model-free reinforcement learning approach that builds a Q-table $Q(s, a)$ of discrete state-action values, useful for learning optimal coverage-maximising schedules on energy harvesting devices [4, 1, 12]. Q-values are the estimated total discounted reward from taking action a in state s and then following the optimal, learned policy π. The discounted reward signal is used to update the Q-table:

$$Q_\pi(s, a) = r_0 + \gamma max_a Q_\pi(\acute{s}, a) \tag{1}$$

γ is a discount factor that balances the importance of immediate versus future reward. Q-learning inference follows a greedy or ϵ-greedy policy, which for a given state picks the maximum Q-value action with probability 1-ϵ and a random action otherwise. Given device resources are limited, and the scheduler should incur minimal overhead, Q-learning's low memory requirements and $\mathcal{O}(n)$ inference and update are fitting. With hourly periods and K = 7 possible actions, the resulting 24x7 Q-table is only 3KB, meaning multiple Q-tables can be used on one device for the scheduling of different connected sensors.

We discretize state into t periods of fixed activation rate. This can be done in practice based on observed/expected event patterns and imposed operational constraints. We also use K possible activation frequencies based on the observed/expected range of event intervals and durations. Our reward for period t is the difference between the number of positive activations N_{p_t} (i.e. activations on which an event is detected) and a weighted sum of negative activations $w_1 \cdot N_{n_t}$. The weighting w_1 adjusts for scale discrepancies, varying detection priorities, and can be period-specific based on expected event patterns.

Q-learning inference and Q-table updates are implemented on-device to show their negligible overhead. We record continuously for 0.1s upon activation to detect events; if an event is detected, recording persists until the event concludes, otherwise the device returns to deep-sleep until its next scheduled activation. Goertzel filtering [17] is used for event detection, processing a 0.1s recording (at 16000kHz) with just 0.03s latency, minimally impacting our device's lifetime. However, Goertzel, and other filtering-based approaches detect any event within their range of covered frequencies, while missing off-frequency ones, potentially resulting in degraded fine-tuning performance. We therefore also implement an alternative integer-quantized one-layer convolutional model for event detection, using ESP-TFLite-Micro. This model is built on spectrograms of 0.1s slices from our Borneo recordings, with event labels generated using BirdNET, and reaches ~70% accuracy on unseen event data; however, no validation is done on the positive slices. ESP-DSP is used for extracting spectrograms from buffered recordings on-device.

The schedule is learned over 24 hours of detections, and evaluated over the following 24 hours, to imitate day-by-day learning with minimal pre-deployment data. We compare our scheduler to a number of fixed baselines that activate the device every n seconds, irrespective of the current period. These baselines are commonly used in real-world deployments, alongside continuous periodic recording of fixed duration [6]. More complex algorithmic schedules (e.g. based on known event patterns) could also be implemented; although these are non-adaptable, they can be used for Q-table initialization to accelerate convergence.

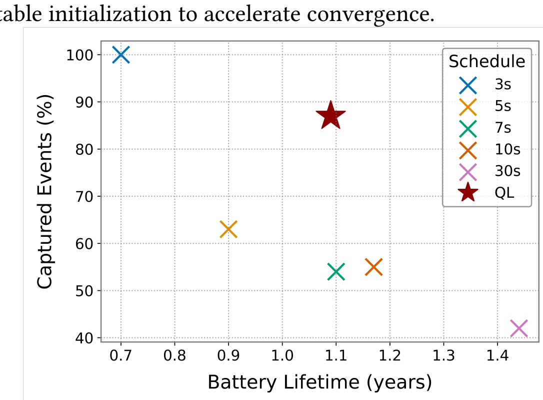

Figure 1: Fixed schedules vs. Q-learning Performance and Battery Lifetime. The device's lifetime is calculated using a 13400mAh LiPo/Li-Ion battery. We model the average operating current based on power measurements of the various device modes, including Q-learning/TFLite inference and update, and the number of detected events. We assume that each detected event triggers a camera activation, resulting in a much higher activation rate than observed for bird/mammal detection in the Sabah rainforest [10], followed by a data transmission. We also do not parallelize operations. This gives us an overall relatively conservative estimate of battery lifetime.

The figure above summarizes the scheduling outcomes. We capture a high percentage of events (85.3% vs ~53.5%) at much reduced power consumption, extending the device's lifetime from approximately 0.77 years on a fixed schedule to over a year (1.07 years). Using TFLite for event detection yields similar results (1.05 years). With a renewable power source for recharging, our lifetime bottleneck moves from power consumption to device/network resilience.

3 Talk Outline

This note outlines a general event-driven scheduler for isolated devices. However, comprehensive biodiversity monitoring necessitates large-scale deployments of networked sensors, prompting further optimization through coordinated scheduling.

We are now exploring a uniquely decentralized, network-centric approach, focusing on scheduling using ordered device proximity groups. We'll cover in our talk how this approach should extend network lifespan as more devices are deployed by minimizing redundancy and distributing resource usage. This approach should also enhance network resilience, as devices can dynamically adjust their schedules if a neighbor fails. We'll also detail how, given a sufficiently dense network, devices in proximity can cooperatively activate each other based on forecasted event patterns/locations, increasing responsiveness for out-of-schedule and non-stationary events.

This effort underpins Terracorder – a uniquely low-power, affordable multi-sensor device for biodiversity monitoring – and facilitates large-scale gathering of rich data essential for supporting conservation initiatives.

References

[1] Fayçal Ait Aoudia et al. 2018. RLMan: An Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks. *IEEE Transactions on Green Communications and Networking*, 2, 2, 408–417.

[2] Rachel T. Buxton et al. 2018. Pairing camera traps and acoustic recorders to monitor the ecological impact of human disturbance. *Global Ecology and Conservation*, 16, e00493.

[3] Vincent Droissart et al. 2021. PICT: A low-cost, modular, open-source camera trap system to study plant–insect interactions. *Methods in Ecology and Evolution*, 12, 8, 1389–1396.

[4] Francesco Fraternali et al. 2020. ACES: Automatic Configuration of Energy Harvesting Sensors with Reinforcement Learning. *ACM Transactions on Sensor Networks*, 16, 4, 1–31.

[5] Alaa Hafez et al. 2023. An Optimized Radar Sensors Schedule Using Genetic Algorithms. In *2023 International Telecommunications Conference (ITC-Egypt)*, 79–82.

[6] Andrew P. Hill et al. 2018. AudioMoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods in Ecology and Evolution*, 9, 5, 1199–1211.

[7] Seunghyeok Jeon et al. 2023. HarvNet: Resource-Optimized Operation of Multi-Exit Deep Neural Networks on Energy Harvesting Devices. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. Association for Computing Machinery, Helsinki, Finland, 42–55.

[8] Bo Jiang et al. 2013. Probability-Based Prediction and Sleep Scheduling for Energy-Efficient Target Tracking in Sensor Networks. *IEEE Transactions on Mobile Computing*, 12, 4, 735–747.

[9] Stefan Kahl et al. 2021. BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, 61, 101236.

[10] Lauretta Andrew Laneng et al. 2021. Camera-trapping assessment of terrestrial mammals and birds in rehabilitated forest in INIKEA Project Area, Sabah, Malaysian Borneo. *Landscape and Ecological Engineering*, 17, 2, 135–146.

[11] James Long et al. 2020. Collaborative duty cycling strategies in Energy Harvesting Sensor Networks. (2020).

[12] Yubo Luo et al. 2021. SmartON: Just-in-Time Active Event Detection on Energy Harvesting Systems. (2021).

[13] Di Tian et al. 2002. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. Association for Computing Machinery, Atlanta, Georgia, USA, 32–41.

[14] J. Wolfgang Wägele et al. 2024. Weather stations for biodiversity: a comprehensive approach to an automated and modular monitoring system. *Advanced Books*, 1, Advanced Books.

[15] Christopher J. C. H. Watkins et al. 1992. Q-learning. *Machine Learning*, 8, 3, 279–292.

[16] Connor M. Wood et al. 2024. Guidelines for appropriate use of BirdNET scores and other detector outputs. *Journal of Ornithology*, 165, 3, 777–782.

[17] Filip Záplata et al. 2014. Using the Goertzel algorithm as a filter. In *2014 24th International Conference Radioelektronika*, 1–3.

[18] Dirk Zeuss et al. 2024. Nature 4.0: A networked sensor system for integrated biodiversity monitoring. *Global Change Biology*, 30, 1, e17056.