Rapid Scaling of Environmental Sensor Networks with Mini-Nodes

Cole Deck
Illinois Institute of Technology
United States
cdeck@hawk.iit.edu

Jack Harrison Mohr
Illinois Institute of Technology
United States
jmohr1@hawk.iit.edu

Abstract

Environmental sensor networks have the potential to augment scientific research and help address urban challenges, but scalability remains a major hurdle. The SAGE network[1], led by Argonne National Laboratory, is a national-scale network of environmental sensors that is spearheading the development of such networks. However, with just over a hundred nodes deployed, scalability remains a challenge due to the high cost of each node. In this paper, we propose a solution to scale the SAGE network by wirelessly connecting each SAGE node with dozens to hundreds of low-cost mini-nodes located nearby. We present a proof-of-concept extension of SAGE's wireless sensor platform, Waggle, that enables the connection of an arbitrary number of additional sensor nodes, which can be built and deployed at a fraction of the cost of a full-scale node. We prototype the detection and mapping of potholes in Chicago as an example use case for a scaled sensor network.

Keywords: IoT, Environmental Sensor Networks, edge computing

1 Introduction

The potential impact of mature, large-scale environmental sensor networks and the internet of things (IoT) has inspired cross-disciplinary enthusiasm for decades. For example, in 2005, Zhao and Brown[15] captured the feeling for IoT's potential following their successful application of a sensor network for monitoring birds on an island:

> "The experiment on Great Duck Island is a small lens into an expansive future. To grasp what might happen, multiply these 190 sensors by 10 million or 100 million and distribute them globally. When the sensor grid becomes ubiquitous, it becomes like an enormous digital retina stretched over the surface of the planet. This planet-scale system could help us understand and address tomorrow's environmental challenges, ranging from monitoring global biodiversity to sensing millions of low-level, non-point sources of pollution."

The nationally-funded development of the SAGE network is one of the current research projects coming closest to making large-scale environmental sensor networks a reality. The SAGE network serves both as a platform for developing the next generation of environmental sensor networks, and as a live open-access tool for scientific research, with experimental applications including environmental sciences [17] [13] [7] and the study of urban dynamics like traffic flow and disease transmission [5] [9] [16].

As is the case with technology in general [8][20], the development of IoT is a conversation between science and commerce. The scaling of the technology requires the funding of government or private industry, and ultimately, the support of civilians. Previous work has affirmed this observation for the IoT domain [12]. One of the early successes of the SAGE project was the SAGE team's focus on getting community input when building the network, including having open panels about privacy concerns, and inviting students to help design the node exteriors[6]. However, a key challenge to massive scaling is to create use cases that warrant funding and to implement the systems in a cost-effective way. While the SAGE system presents a small-scale proof of concept, a key question that remains is how the network will be scaled.

The focus of this paper is to demonstrate that one solution to the scaling problem is to augment each SAGE node with dozens or hundreds of low-cost mini-nodes. This strategy has two advantages: lower cost and increased capabilities. The low cost of the mini-nodes makes it easier for local governments to back the project, and the resulting increase in capabilities increases the justification for funding.

Dr. Dan Reed, a collaborator on the SAGE project puts the cost of each SAGE node at approximately $15,000 [18]. He notes that mini-nodes could be produced at scale for approximately $100 each (though the actual cost would also factor in installation, maintenance and the cost of an appropriately durable casing). As a back-of-the-envelope calculation, to place a sensor node every 100 meters in a 10 square kilometer area centered in Chicago's downtown Loop area, you would need 100,000 nodes. This level of node-density could enable many useful applications. However, to use only the existing type of SAGE nodes would cost approximately $1.5 billion in node hardware (not including installation). In contrast, using mini-nodes would cost approximately $10 million. To put this into perspective, light poles can cost the city a few thousand dollars each and are more densely placed[11].

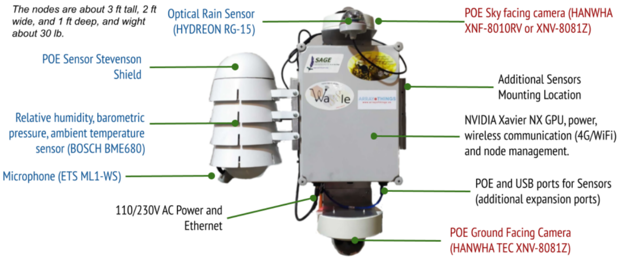

Figure 1. Wild Waggle Node breakdown[2]

Our core hypothesis is that the combination of demonstrated public usefulness and low-cost implementation will be critical factors in the successful evolution of large scale environmental sensor networks like SAGE. We therefore demonstrate how SAGE's Waggle platform can be modified to receive additional sensors from mini-nodes, and offer an example of a scaled-up network's potential public usefulness by prototyping a pothole detection system for the city of Chicago.

2 Design

2.1 Background

Our work builds on the Waggle platform [4], which is designed to address some of the challenges posed by large open-access environmental sensor networks like SAGE. SAGE and Waggle already make sensor networks more available and affordable than prior attempts by leveraging edge computing and lower-cost sensors.

2.1.1 Wild Waggle Nodes. A Wild Waggle Node is a fully featured Linux sensor hub that is weatherproof and equipped with a variety of sensors and cameras. It contains hardware capable of performing AI workloads, allowing for privacy-oriented edge computing. While it is more affordable than other existing approaches, it is still in the range of $15,000 US dollars per node due to the high end hardware and sensor array. For reference, each node contains 1-2 NVIDIA Xavier NX compute modules, which typically cost $499 USD each. See Figure 1 for a high level overview of the hardware present.

Wild Waggle Nodes connect to the internet and are able to download apps from the Edge Code Repository (ECR) which is an app catalog for various machine learning and AI analysis plugins that run on each node using the different sensors as inputs.

2.1.2 Waggle Architecture Overview. Waggle is a platform used to build edge computing applications for environmental sensors and research. Wild Waggle Nodes run an embedded linux distribution using k3s to support a runtime capable of allowing the use of plugins, which are mainly used for analyzing sensor data to provide a more basic output, e.g. object detection, bird identification, and many more. Plugins can be downloaded from the ECR onto individual nodes, which then run plugins locally using one or more NVIDIA Xavier NX compute modules for accelerated ML and AI. Plugins are essentially python programs running in docker containers, and send and recieve data from the Waggle platform using the pywaggle API library.

While plugins are mainly intended to take sensor data from the Waggle API, perform machine learning workloads, and return data back to waggle, there is really no limit on what a plugin could actually do. What is very helpful is that outputs from plugins are considered to be software-defined sensors and can be accessed by any other plugin just like physical sensors. Because of this freedom, plugins must be manually approved before they are added to the ECR, though this creates the opportunity to use the plugin system in ways unintended by the designers.

Mini-nodes communicate over MQTT[14], a popular IoT optimized messaging protocol. In order to connect mini-nodes to the Waggle Edge Stack, we can run a python program using the plugin system that offers a MQTT broker and client, which listens for mini-node connections over MQTT based on a configuration file, and forwards data to the Waggle API. Waggle plugins are (officially) allowed to send & receive data to and from other plugins, so it is trivial for a typical Waggle machine learning plugin to acquire data from our "bridge" plugin.

2.2 MQTT Waggle Bridge Plugin

The MQTT Waggle Bridge Plugin is a small python program inside of a docker/k3s container. It can take arbitrary sensor data from a python data structure and pass it through to the Waggle Plugin API untouched, with some additional metadata to indicate the source of the data.

When the python application first runs, it uses the `asyncio` library to spin up a MQTT broker in the background, and subscribes to pre-defined sensor topics with a MQTT client. Topics are defined in a yaml configuration file. The structure of the yaml file is used to create a hierarchical sensor topic structure dynamically. For example, this is a valid configuration that defines a sub-group of nodes:

```
sensors:
  pi1:
    - "camera"
    - "uptime"
  subnet1:
    pi2:
      - "microphone"
      - "uptime"
    pico1:
      - "soil"
      - "uptime"
```

This allows for the creation of categories, networks, and more, without any changes needed outside of the configuration file. When data is relayed to Waggle, the innermost section ("soil", "camera", etc) are used as the sensor type. This means that, for example, all camera streams would get published to
`network.bridge.sensor.camera`. In order to determine *which* camera stream was published, the metadata of the message contains a `sensorID` which is a unique ID for each sensor, a `deviceID` which is a unique ID for each mini-node, and the full hierarchy of said sensor from the configuration. A similar approach is already used in the Waggle platform, as many Wild Waggle Nodes already contain 2 cameras. `sensorID` and `deviceID` are auto-generated based on the configuration file, but a mini-node can specify these values if desired.

Sensors that are not defined in the bridge configuration file will be ignored, as the server will only listen on the topics that are defined.

2.3 Mini-nodes

In contrast to prior work[19], mini-nodes have no specific hardware requirements, as long as they can connect over IP to a full Waggle node, and support a MQTT client. In our testing, mini-nodes were tested on VM, Raspberry Pi 3, Raspberry Pi Pico W, and ESP32 platforms. These are very affordable options, starting at less than $6 for a Pi Pico W. Existing IoT sensor firmware that uses MQTT such as Tasmota on an ESP32 can be used as a mini-node with very small changes made to the bridge server or firmware, simply to match up the json fields used.

Mini-nodes do not need the Waggle API libraries in their code. This is very beneficial as the waggle dependencies include everything needed to run ML workloads, which, when built into a docker image, requires several gigabytes of disk space. The only place the Waggle API is required is in the Bridge plugin itself.

Data sent by a mini-node is a very simple JSON structure. The only required field is `data` which contains the sensor data itself, and it must be published to the full topic that matches that sensor on the server configuration. Optional fields include `timestamp`, so the mini-node can report exactly when the data was collected. If it is not included, the bridge plugin will input the time it receives the data, as Waggle requires a timestamp. Mini-nodes can also specify a `sensorID` and/or a `deviceID`, as long as it isn't one of the autogenerated IDs. Autogenerated IDs start at 0, so if explicitly set IDs are to be used, starting at a higher number than the amount of sensors/devices is all that is required.

2.4 Mini-node Communication and Power

As the only requirement is that a MQTT message can make it's way to the bridge server, there are many ways to connect sensors to a Wild Waggle Node. A good approach would be to use a wireless communication protocol that ideally supports meshing, such as Zigbee, so sensors can be at a significant distance from the main Waggle node while retaining the ability to send data. The benefits of this approach include the lack of any physical data lines for ease of installation and flexible placement options. A protocol like LoRaWAN could also be useful, allowing connections at a much higher distance.

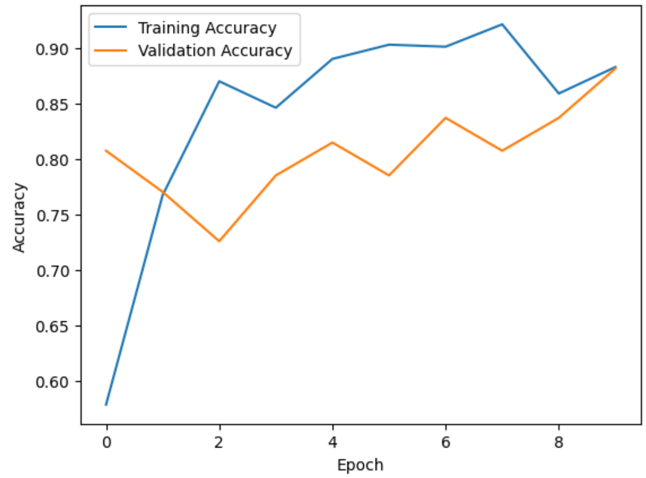

Figure 2. Pothole classifier accuracy

MQTT messages do not need to directly reach the bridge from the source mini-node, as long as the message arrives to the correct topic then there is no difference in functionality.

Since mini-nodes can be very lightweight, low-power microcontrollers, they could also be solar or wind powered, so in addition to wireless communication, this allows for mini-nodes to be completely independent of any external wiring, so installation can be as simple as physically placing a complete node in a desired location. In comparison to a Wild Waggle Node, which requires AC 110/230 V power, mini-nodes are potentially much quicker and easier to setup.

3 Evaluation

3.1 Demo Application: Pothole detection

One potential use case for a scaled up sensor network in an urban environment is pothole detection. Having mini-nodes on every street could enable the real-time monitoring of road conditions, augmenting a city's ability to repair roads and monitor infrastructure. The following describes our process.

First, we train a convolutional neural network for binary image-level classification of street images as either containing, or not containing potholes. Our model has an approximately 87% validation accuracy (see Figure 2 and 3). Accuracy could surely be improved with more tweaking.

Next, we simulate the images that would be captured by the mini-nodes by using Google Maps street view images through the Maps API. We sample only a few blocks

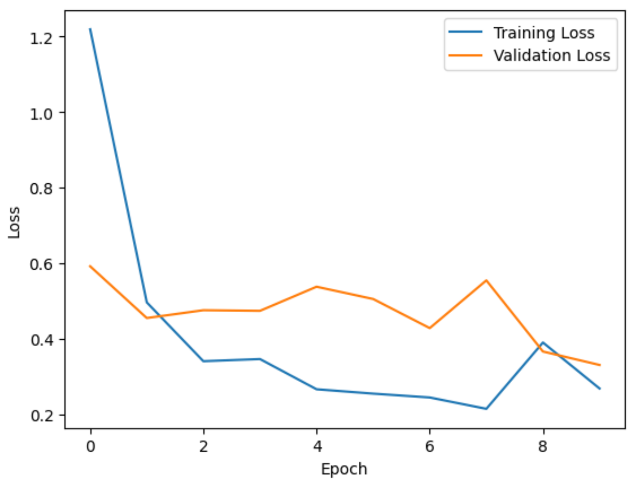

Figure 3. Pothole classifier Loss

in downtown Chicago because of computational and budget constraints. We then classify each of these images using our pothole classifier. Finally, we check to see if the detected potholes have been filled by the city. The City of Chicago maintains an up-to-date, publicly accessible dataset of all the potholes filled by the city. Furthermore, both the Google Images and the City of Chicago pothole data entries are tagged with geographic coordinates, so it is possible to check for each detected pothole whether any potholes in the same location were filled since the date of the image capture.

Finally, we display the images in a user-friendly interactive map (Figure 4) using the Folium Python library. The map references the images by URL and is easily portable to the web, making it simple to set up the user interface as a real-time interactive web dashboard accessible both the city officials and to the public. The map also allows users to click on 'pinned' pothole locations to see an image of the pothole. Thus, humans can verify the machine's inference before heading out to fix the potholes.

Currently, civilians in Chicago are expected to call the non-emergency 311 number to notify the city that a pothole needs to be filled. While the City of Chicago's pothole dataset suggests that the city has an excellent track record of responding to requests, anyone who has been on the streets of Chicago will tell you that potholes remain numerous. Therefore, a plausible but unconfirmed hypothesis is that the bottleneck is one of information. That is, perhaps the city is not aware of the existence of specific potholes and people do not call in many cases. Our prototype thus demonstrates that the large-scale mini-node augmentation of the SAGE network could be a solution to Chicago's pothole problem by fixing the informational bottleneck of pothole detection. It would not be plausible to solve this problem with the current SAGE nodes because the cost of putting multiple nodes on every street would be prohibitive.

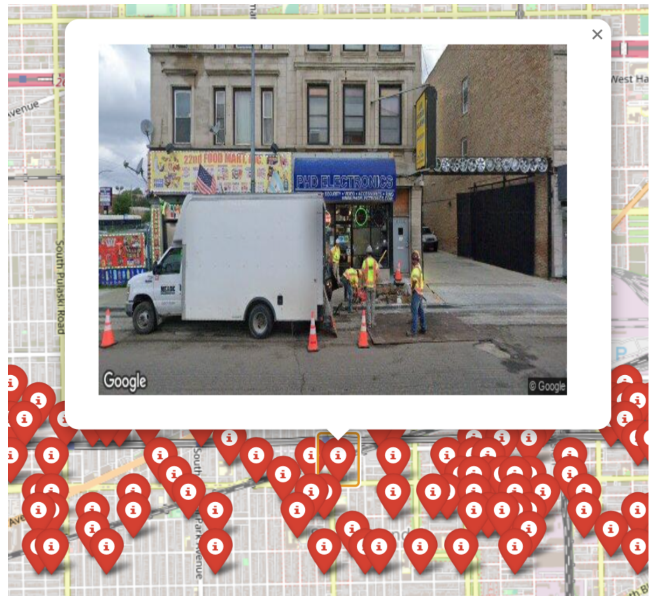

Figure 4. pothole map UI with image popups

On the other hand, this problem could be solved in still more cost effective ways. For example, the City of Chicago could survey the roads each month with a drone and use that as input to the pothole detector. Similarly, the data could be crowd-sourced from cars containing cameras. Both of these solutions are far cheaper and faster to implement than a large network of mini-nodes. The existence of these alternative solutions to Chicago's pothole problem reflects the importance of continuously questioning the framing of the problem statement, especially when proposing high-cost solutions. Therefore, to justify the expense of the mini-nodes, the mini-nodes would need to offer many additional use cases that bring value to the city. Thanks to Waggle's ECR, there are already many existing use-cases that can be expanded to mini-nodes from bird identification to monitoring traffic. Given a sufficiently useful set of applications it is plausible that local governments might choose to partner with SAGE to install dense arrays of mini-nodes in urban areas.

4 Limitations and Future Research

While this project is able to successfully transmit mini-node sensor data into the Waggle platform, additional work is needed to better integrate into Waggle. existing ECR apps will need to make very slight changes to their code to take inputs from the Bridge Plugin, and this requirement could be removed by having mini-node support added directly to Waggle's core software or some work to further integrate the Bridge Plugin. Messaging protocols other than MQTT could be explored, to optimize for different data types or networks.

5 Related Work

In a recent survey of low-cost sensor networks, Mao et al.[12] highlight that the foremost challenges to large-scale sensor network deployment are "non-technical factors such as stakeholder engagements, socio-economic contexts, financial and operational mechanisms." This observation is afforded by the decades-long research efforts to develop appropriate network protocols, hardware design, and node architectures that make large sensor networks possible[10].

There have been prior attempts to extend the Waggle platform with smaller, lower cost nodes such as micro-Waggle[19], however there are several limitations with micro-Waggle's approach; it only supports Particle.io devices, which are pretty affordable, but they also require a direct connection to the internet to access the Particle.io cloud, and data is not sent directly to a Waggle node[3]. Opening up to the internet also increases attack vectors and restricts connectivity options.

6 Conclusion

In this paper, we propose the use of low-cost mini-nodes as a solution to scaling environmental sensor networks. We modify the SAGE network's Waggle platform to accommodate an arbitrary number of mini-nodes. While the SAGE network has already demonstrated useful scientific applications, the future impact of large-scale environmental sensor networks will depend in part on the economic viability of scaling these systems by orders of magnitude.

References

[1] 2020. SAGE Continuum. Website. https://sagecontinuum.org/ Accessed: May 4, 2023.

[2] 2022. GETTING STARTED - WILD SAGE NODE – VERSION 1. PDF file. https://docs.waggle-edge.ai/assets/files/WSN_GS_V1-1308f536b647df844738f6f10f0db1f6.pdf Accessed: May 4, 2023.

[3] Kenan Arica. 2019. README.md of micro-Waggle. GitHub repository. https://github.com/waggle-sensor/microWaggle Accessed: May 4, 2023.

[4] Pete Beckman, Rajesh Sankaran, Charlie Catlett, Nicola Ferrier, Robert Jacob, and Michael Papka. 2016. Waggle: An open sensor platform for edge computing. In *2016 IEEE SENSORS*. IEEE, 1–3.

[5] Charlie Catlett, Pete Beckman, Nicola Ferrier, Howard Nusbaum, Michael E. Papka, Marc G. Berman, and Rajesh Sankaran. 2020. Measuring Cities with Software-Defined Sensors. *Journal of Social Computing* 1, 1 (2020), 14–27. https://doi.org/10.23919/JSC.2020.0003

[6] Charlie Catlett, Pete Beckman, Nicola Ferrier, Michael E Papka, Rajesh Sankaran, Jeff Solin, Valerie Taylor, Douglas Pancoast, and Daniel Reed. 2022. Hands-on computer science: the array of things experimental urban instrument. *Computing in Science & Engineering* 24, 1 (2022), 57–63.

[7] Dario Dematties, Bhupendra A Raut, Seongha Park, Robert C Jackson, Sean Shahkarami, Yongho Kim, Rajesh Sankaran, Pete Beckman, Scott M Collis, and Nicola Ferrier. 2023. Let's Unleash the Network Judgment: A Self-Supervised Approach for Cloud Image Analysis. *Artificial Intelligence for the Earth Systems* (2023), 1–34.

[8] Yuval Noah Harari. 2014. *Sapiens: A brief history of humankind*. Random House, Chapter The Wheels of Industry, 334.

[9] Hazel Han. [n. d.]. Vehicle Tracking. https://sagecontinuum.org/science/vehicle-tracking [Online; accessed 4-May-2023].

[10] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. 2002. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications* 1, 4 (2002), 660–670.

[11] Lightmart contributors. [n. d.]. How much do street lightpoles cost? https://www.lightmart.com/blog/how-much-do-street-light-poles-cost/ [Online; accessed 4-May-2023].

[12] Feng Mao, Kieran Khamis, Stefan Krause, Julian Clark, and David M Hannah. 2019. Low-cost environmental sensor networks: Recent advances and future directions. *Frontiers in Earth Science* 7 (2019), 221.

[13] Enrique Nueve, Robert Jackson, Rajesh Sankaran, Nicola Ferrier, and Scott Collis. 2021. WeatherNet: Nowcasting Net Radiation at the Edge. In *2021 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE, 1–7.

[14] OASIS. 2019. MQTT Version 5.0. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html. Accessed: May 4, 2023.

[15] Robert L Olson, David Rejeski, et al. 2005. *Environmentalism & the technologies of tomorrow*. Island Press.

[16] Ori Zur. [n. d.]. Social Distancing. https://sagecontinuum.org/science/social-distancing [Online; accessed 4-May-2023].

[17] B. A. Raut, P. Muradyan, R. Sankaran, R. C. Jackson, S. Park, S. A. Shahkarami, D. Dematties, Y. Kim, J. Swantek, N. Conrad, W. Gerlach, S. Shemyakin, P. Beckman, N. J. Ferrier, and S. M. Collis. 2023. Optimizing cloud motion estimation on the edge with phase correlation and optical flow. *Atmospheric Measurement Techniques* 16, 5 (2023), 1195–1209. https://doi.org/10.5194/amt-16-1195-2023

[18] Dan Reed. 2022. Cyberinfrastructure: Lots of Room at the Edge. Blog post. https://www.hpcdan.org/reeds_ruminations/2022/06/cyberinfrastructure-lots-of-room-at-the-edge.html Accessed: May 4, 2023.

[19] V. A. Rivera, S. Shahkarami, R. Sankaran, W. M. Miller, A. I. Packman, and P. H. Beckman. 2018. micro-Waggle: Low-profile, modular sensor platform for environmental research. In *AGU Fall Meeting Abstracts*, Vol. 2018. Article H13P-1994, H13P-1994 pages.

[20] Lynne G Zucker and Michael R Darby. 2007. Virtuous circles in science and commerce. *Papers in Regional Science* 86, 3 (2007), 445–470.