

An Easily Deployable Wireless Imaging System

John Hicks[†] Jeongyeup Paek[§] Sharon Coe[‡] Ramesh Govindan^{*} Deborah Estrin[¶]

Center for Embedded Networked Sensing

^{†¶}University of California, Los Angeles ^{§*}University of Southern California [‡]University of California, Riverside
[†]johnhicks@gmail.com, [§]jpaek@usc.edu, [‡]sharon.coe@ucr.edu, ^{*}ramesh@usc.edu, [¶]destrin@cs.ucla.edu

Abstract

We report on the development and deployment of a wireless, image-based environmental monitoring system. Our system uses low-power Cyclops cameras and the Tenet general-purpose sensing system. It leverages Tenet's built-in support for reliable delivery of high data-rate sensing data, and its flexible scripting language that enables mote-side image compression. Our three-month long deployment at the James San Jacinto Mountain Reserve resulted in over 102173 images collected from a 19-node network deployed over an area of 0.05 square miles, despite highly variable individual node availability. Our biologist users found the on-line, near-real-time access to images to be useful for obtaining data on the nesting behavior of bird species.

Categories and Subject Descriptors

C.3 [Special-Purpose and Application-Based Systems]: real-time and embedded systems, microprocessor/microcomputer applications

General Terms

Design, Experimentation, Performance, Reliability

Keywords

Sensor Networks, Imaging, Habitat Monitoring, Deployment

1 Introduction

Studies have been taking place for several years at the James San Jacinto Mountains Reserve on the breeding biology of species of birds that normally build their nests in natural holes in trees. For such species, researchers often place wooden boxes, called *nest boxes*, on trees that birds can use for nesting because they allow the researcher to view the contents of the nest more readily compared to a cavity within a tree. Nest boxes have been placed around the reserve and have been occupied by breeding birds. However, observing the day-to-day changes of breeding behavior in these boxes is extremely labor intensive. Each box must be checked daily by a biologist in the field, or it must be wired with a permanent camera for remote observation.

Thirteen of the boxes at the reserve have indeed been wired to give high quality, high data-rate and high resolution images, but they have limitations. The number of wired boxes is low as they are restricted to locations near sources of power and Ethernet. Also, some of the wired boxes have never been used by birds for nesting and hence not provided

useful biological data, but since the labor required to move the cameras to occupied boxes is high, they have remained where originally established.

To address these issues, we need an imaging system that can be deployed easily and moved flexibly over a large area at places where wired infrastructures are less available. The system should support a large number of cameras and should operate for long periods of time without excessive maintenance. Furthermore, it needs to be easy to use by non-engineers.

In this paper, we devised and implemented a low-power, scalable, wireless imaging system using off-the-shelf hardware and Tenet, a readily available open-source software package for programming wireless sensor networks [2]. Low-power sensor networks are not a new theme, and there already exist various systems that can take environmental measurements and return them in real-time. However, there are few such systems for low-power, high data rate retrieval of images, which are also configurable and usable by those with little knowledge of embedded programming. The large size and inherent complexity of images along with a large number of cameras over a wide area gives rise to routing, reliability, congestion control, power, and other issues not seen in lower data rate systems. One known solution stems from a similar deployment at *James Reserve* [1]. Our deployment uses the same hardware, including the same nest boxes and their locations, but implements a completely revamped software system. We chose the standardized Tenet package which gives a number of usability and flexibility advantages outlined in Sections 2 and 3.

2 Deployment

Here we describe our deployment at *James Reserve* including motivation, requirements, our approach, and hardware setup.

2.1 Motivation

Avian ecologists study the behavior of birds during the nesting season to answer biological questions that relate to the laying, incubation, and hatching of eggs, and the survival of nestling birds. The breeding season can last up to three months during Spring. Usually a minimum of 30 nests of a single species of bird is needed to provide a statistically robust analysis.¹ Each of these nest boxes needs to be checked

¹To achieve this, a researcher studying birds in nest boxes would typically place twice that number of nest boxes in the field to give

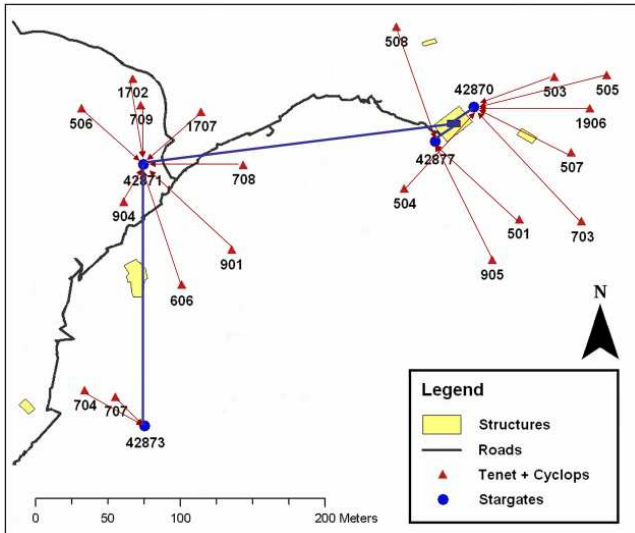


Figure 1. A map detailing the location and topology of our deployment at *James Reserve*.

manually on a regular basis to determine which are occupied by birds, and then monitored regularly (in some cases, daily) to take data on the breeding behavior of interest. Such efforts can require a considerable investment of time and labor. The low rate of data sampling makes answering many scientific questions infeasible. In addition, manually opening and inspecting the nest boxes to obtain data can create a disturbance that can result in adult birds abandoning their nest.

To this end, the goal of our deployment is to demonstrate a system that can be used by ecologists to continuously observe the interior of nest boxes spread over a large area for the duration of the breeding season with minimal disturbance to the breeding birds.

2.2 Requirements

There are three main requirements for our system. First is the ease of use by non-engineers, both in set-up and running/maintaining of the deployment. The system must be easy to use by biologists with minimal dependence on computer programmers. The system should allow the user to easily add and adjust parameters. Also, we need an end-to-end system to allow the operator to monitor incoming images and data in real time. This will allow the personnel who are maintaining the system to detect problems as they happen and to alter various settings to maximize data return.

Second, the system should not miss important events. This means that images should be taken at a relatively high rate, high enough to detect events such as bird presence/absence with high accuracy.

Finally, the system must observe enough birds during their breeding period to derive statistically robust results. This means that the system should be scalable to acquire and manage images from a large number of nest boxes spread over a large area.

ample “choice” to birds in selecting nest sites, with the hope of achieving 50% occupancy.

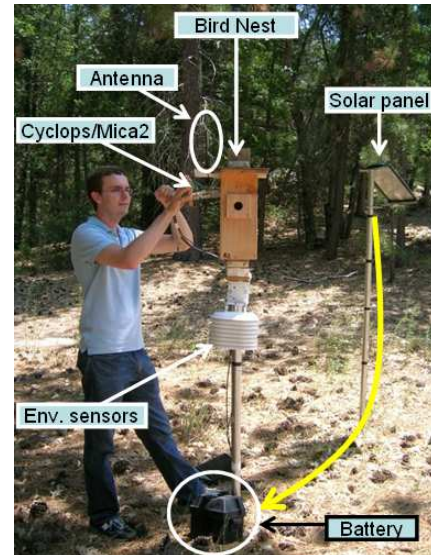


Figure 2. The outside of a typical nest box. Notice the solar panel mounted at a distance for optimal sunlight, the nest box mounted on a pole to avoid predation, and the shelf at the top holding the hardware.

In addition to the above requirements, the incremental cost of adding more nest boxes to the system should not be too high. Also, each camera must be low power to reduce maintenance costs, or, with the help of solar panels, eliminate those costs.

2.3 Our Approach

The Tenet software system includes all necessities for basic wireless sensor network programming: drivers, routing protocol, flow control, end-to-end reliability, a two-tier network hierarchy, and a simple scripting language for easy programming of applications. These properties, in addition to Tenet’s flexibility towards future improvements, make Tenet the ideal choice for our deployment. We have added new software pieces to Tenet for using Cyclops camera and MDA300 sensor board, implemented basic image compression, and formed a complete end-to-end system.

The nest boxes are generally sparse, being spread 50 to 100 meters apart in areas of dense trees and foliage. Newer 2.4 GHz radios tend to perform poorly in these environments; instead we used Mica2s with 433 MHz radios as our wireless communication hardware in the mote-tier. This drastically limits available bandwidth but propagates farther through the foliage.

These factors together enabled us to reach our goal of increasing the temporal resolution of data, providing near real-time monitoring which saved time and labor for the biologist, and allowing a larger spatial area to be covered with sensors.

2.4 Hardware Setup

Here we describe the specifics of the hardware and an overview of the configurations used in the deployment. We have one Linux server machine running in the *James Reserve* server room. The server communicates with four Stargates, which have been placed around *James Reserve* and together

constitute the upper tier of the network. The server and the Stargates are connected via Ethernet or 802.11. Each Stargate is connected to a Mica2 433 MHz mote which all have 8.5 dBi omnidirectional antennas. These Mica2s are used to communicate with nearby Cyclops nodes, which comprise the low power, lower tier of the network. This network topology is detailed in Figure 1. While our system does support multihop routing (and we did log some temporary multihop paths), nest box placement was determined from a previous, single-hop deployment which, when combined with the large Stargate antennas, eliminated most multihop route formations [1].

Each sensor node contains a nest box, power infrastructure, and the embedded hardware necessary for imaging, environmental sensing, and communication. The wooden nest boxes are all custom-made and contain a removable shelf which holds the Mica2 mote for communication, a Cyclops camera for imaging [4], and a MDA300 board for environmental sensing.² Figure 2 shows the outside of a nest box.

The shelf has a clear plastic bottom through which the camera takes images while protecting the hardware from dirt and disturbance by nesting birds. The power infrastructure is a medium sized 12V sealed lead acid battery continually charged by a solar panel. For most nodes, the solar panel provides enough power for unlimited node up-time.³ Finally, while most nodes had small 433 MHz dongle antennas mounted on the top, a few nodes with extremely poor connectivity required 9 dBi directional antennas pointed at the nearest Stargate.

3 System Architecture

The Tenet software collects images and environmental data from every sensor node and stores them on the local server at *James Reserve*. Tenet applications run on the local server, and multiple Stargates act as Tenet-masters which relay commands to and data from the Cyclops nodes. A back-end server at CENS retrieves the data from the local server via Internet, and archives and processes the data. These components together give a complete, real-time, end-to-end system.

3.1 Tenet Advantages

Our bird nest monitoring system has several requirements including forming multi-hop routes over a tiered network, transferring images frequently, end-to-end packet reliability, and congestion control. The flexibility of deploying new nodes and moving unused ones (e.g. cameras in nest boxes not occupied by nesting birds) is also a requirement, as is a convenient way to easily start, stop, modify, and reconfigure the application. These considerations have led us to adopt Tenet software for our system as Tenet already addresses most of above requirements and allows easy addition of new functionality.

²Internal temperature, internal humidity, external temperature, external humidity, and voltage sensors are connected to the MDA300 board.

³On a few nodes, the solar panel did not receive enough solar radiation throughout the day which resulted in failed nodes due to loss of power.

Tenet is a software package for flexibly programming a tiered network of sensors. The Tenet system consists of motes and less-constrained 32-bit platforms called masters. All applications run on the masters and task motes using a generic tasking API that allows the user to run simple programs on the master nodes to configure, control, sample, and process data without having to reprogram the motes. Tenet constructs seamless multi-hop routing over a tiered network of motes and masters which enables flexible deployment of sensors over large area. Tenet also provides end-to-end reliable delivery of packets with built in congestion control capability. Reliable delivery is an application requirement for our system, otherwise image quality can be severely compromised. It also allows our system to use loss-intolerant image compression techniques to increase effective network capacity since these techniques require 100% packet delivery for correct decompression. Congestion control allows our application to adapt its image transfer rate to network scale and wireless environment. By using Tenet, we can reuse all of the above networking and sensor data extraction code, thereby significantly reducing application development time.

3.2 Our Application

The goal of our application is to repeatedly collect from every node a Cyclops image along with MDA300 sensor readings as frequently as possible. Developing this application using Tenet involved adding several pieces of code into Tenet. We ported device drivers for Cyclops camera and MDA300 data acquisition board into Tenet, added Tenet tasklets and its APIs for accessing these devices, and designed an image compression algorithm to reduce the image transfer latency. All of these were fairly straightforward to implement in Tenet, and are now integrated into the Tenet source distribution so that subsequent deployments can reuse this code. Finally, two Tenet applications were written to execute the following tasks in the Tenet scripting language [2]:

```
TASK1:
image_getPackBits(0,40,100,1,16,200,200,5,9)->sendrcrt()

TASK2:
wait(1000) -> mda300(0,1,100,0) -> mda300(0,2,101,0)
-> mda300(0,3,102,0) -> mda300(0,0,103,153)
-> mda300(1,0,104,153) -> send(1)
```

The first task takes a 200x200 resolution black and white image, compresses the image using modified *PackBits* algorithm with a threshold of 5 (described below), fragments it into 40 data bytes per packet, and sends the packets back using the RCRT protocol [3]. The second task reads the five ADC channels on MDA300 board and sends the data back using the packet reliable transport protocol. Then a simple server side script executes the first task, waits for all responses to arrive, executes the second task, and repeats this process indefinitely.

For compression, we implemented a modified *PackBits* algorithm. *PackBits* is a fast, simple loss-less compression scheme for run-length encoding of data originally developed by Apple. This scheme was chosen because the encoding operation at the mote-end is simple and can be done on-the-fly without requiring a full image size buffer. We modified

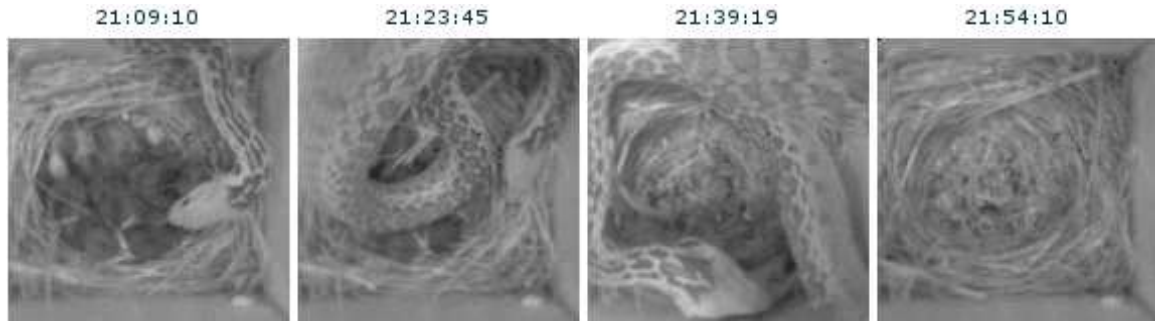


Figure 3. A series of images captured by a Cyclops camera during the post-hatching phase. Notice the snake entering the box and leaving in well under an hour, an event that would be missed with manual, daily observations.

PackBits into a lossy algorithm so that a sequence of values within two thresholds consists a ‘run’.⁴ The threshold value trades-off image quality for increased compression, and setting it to zero is equivalent to using the original loss-less algorithm. We chose a threshold of ‘5’ as a reasonable balance; we achieved around 25% reduction in image size while experiencing little visible degradation in image quality. Finally, the RCRT protocol included in Tenet performed congestion control and rate adaptation as well as loss recovery during the image transfers.

3.3 Back-end Server

The back-end database and image viewer completes our system. We used Sensorbase.org, a public database at CENS which stores environmental data and images. As mentioned, the images and environmental data are collected at the Tenet master node, a Linux server running at *James Reserve*. The data are pulled from this machine and entered into the Sensorbase database at regular, 15-minute intervals. The images and data are finally displayed on one of two flash viewers. One allows easy image browsing while the other is used to monitor the status of the system in real time.

4 System Evaluation

To evaluate our system, we first discuss deployment experiences. Then we give an overview of measured system metrics from two perspectives. We measure sensor node up-time and occupancy, and network behavior.

4.1 Deployment Experience

As much of the deployment hardware was already in existence from previous deployments, most of our work in preparation was spent developing the software using the Tenet system. Our remaining time was spent deploying the hardware with two additional trips to *James Reserve* for routine system maintenance. We deployed eight nodes during a trial run in early May, then deployed the entire 19 node system on May 9th and brought the system up in the late afternoon. The official end date for our deployment was exactly three months later, or Aug 9th at 5 pm.

The server machine at *James Reserve* is connected to the Internet, and thus we were able to remotely monitor and re-program the deployment. We needed to do this several times when either parameter changes were made to the application

⁴In our image data, each byte represents a gray scale value ranging between 0 and 255.

or diagnosis of the network was required. As time went by, there were three distinctive failure events that were observable remotely; 1) unreachable sensor node, 2) sensor node ping-able but not returning images, and 3) rebooted Stargate. The first symptom was usually due to a depleted battery, which in turn was usually due to solar panel being in the shade for extended amount of time. Broken antenna connectors and malfunctioning power systems were other reasons. The second symptom was either due to low battery (not high enough to activate the Cyclops) or a loose connection between the mote and the Cyclops. We are not sure why one of the Stargates rebooted several times during our deployment, but we believe it is may have been due to power outages.⁵

We returned to *James Reserve* twice, once on June 4th and again on July 18th to fix failed nodes and base stations. All repairs were minor and consisted of replacing improperly functioning 12V batteries, reconnecting connections that had come loose, and adding directional antennas to nodes with poor connectivity. These repairs could, in larger deployments, be taken care of by moderately technically-aware domain experts.

In total, our system collected 102173 images from 19 nest-boxes, providing biologists with information about nest-box occupancy and breeding behavior of three species of birds. One highlight of the observations made from the images was that of a snake that was able to enter the nest box consuming nestlings on May 19th. Without these images, using traditional methods of a biologist manually checking nest box contents daily or even less frequently, the reason for the disappearance of nestlings from the nest would have been unknown.

4.2 Node Examination

Figure 4 shows the up-time of the 19 nodes in our system. There was one major system outage between May 21st and June 5th which was due to power failures at *James Reserve*, after which the system did not come up. On June 4th, we returned to the reserve to rectify the issue. We also added 9 dBi directional antennas to two farther out nodes, notably 905 and 703. On July 18, we returned a second time to further fix broken nodes. Batteries were replaced and solar panels adjusted on nodes 1906, 904, and 709. Unfortunately we

⁵*James Reserve* power is supplied from generators and solar panels so there are occasional brownouts and outages.

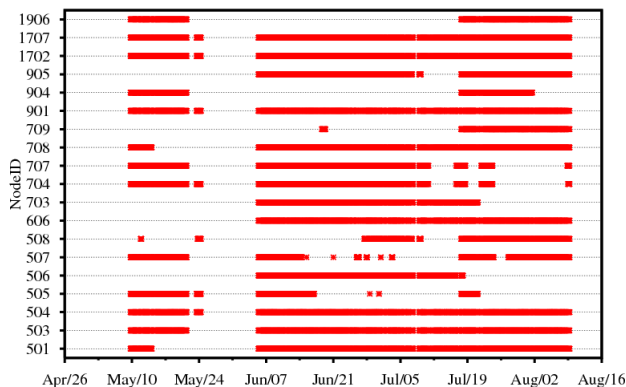


Figure 4. Up-time of all nodes over the season

almost immediately had power outage with Stargate 42873, to which nodes 704 and 707 connect.

We have collected a total of 102173 images. Not counting the 2 week power outage at *James Reserve*, our node up-time ranges from about 90% for our stronger nodes which had no power issues, and down to around 40% for our weaker nodes. 709 was our weakest node, returning very few images until its battery was finally replaced.

4.3 Network Evaluation

As there were several times when the image application was stopped for maintenance or debugging purposes, our networking log files are discontinuous. We present the observations we have made for one week during July 21 – July 28. During this period, there were total of 9940 attempts to transfer an image from a total of 19 nodes. Among these attempts, 1489 were made by three nodes when they had low battery and thus were unable to turn on the camera. Out of 8453 actual image transfer attempts, 79 attempts resulted in incomplete transfers. Hence 8372 image transfers, which corresponds to 99% of initiated transfers, were completed during this one week period. For these complete images, the average data rate achieved by the network was 1.1 packets/sec per node, and the average number of packets required to deliver one image was 833.2.⁶ As a result, one image transfer took 12.6 minutes on average. This was achieved despite extremely poor link connectivity (e.g. node 905 had PRR of less than 50%). Figure 5 shows the end-to-end packet delivery ratio along with their RSSI readings to nearest routing parent node for each node when we tested the network during the deployment using best-effort data delivery without loss recovery.

5 Conclusion

We have demonstrated a complete, scalable, end-to-end imaging system to unobtrusively observe biological phenomena. Our deployment has shown that our system design has met most of the application requirements: ease of use, sufficient image transport rate, scalability exceeding that of wired cameras, and flexibility of deployment. This was made possible by using Tenet.

⁶The size of each image is 40kB, but the number of encoded packets differ since the compression ratio differ for each image.

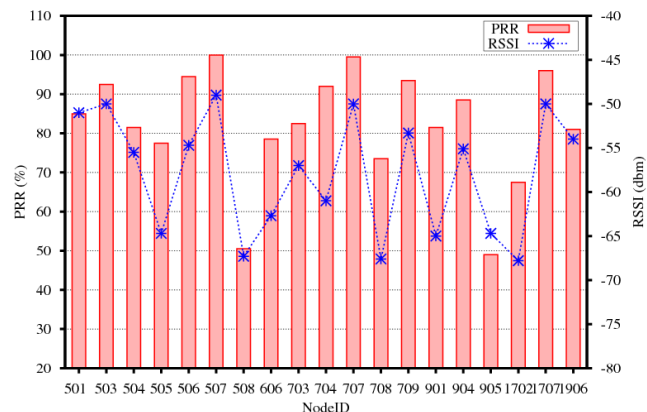


Figure 5. Link connectivity of each node: Packet reception rate and signal strength to next hop node.

Tenet software made it easy to monitor, modify and re-configure the application behavior from a remote location without re-programming the motes. Tenet’s rate-controlled reliable transport along with it’s tiered architecture resulted in an image retrieval rate of approximately 3 images per node per hour. We can further improve this by sending images only when a significant change in image is detected, which will reduce energy consumption and traffic congestion, and thus increase both node up-time and image transfer rate. Future deployments can also benefit from a better compression algorithm or newer-generation radios.

Finally, our system proved to be scalable. Although we have not demonstrated the benefits of multihop routing in the mote-tier due to nest-box placements governed by previous non-Tenet deployment, Tenet’s multihop tiered routing will enable larger network with greater spatial reach.

6 References

- [1] S. Ahmadian, T. Ko, J. Hicks, M. Rahimi, D. Estrin, S. Soatto, and S. Coe. Heartbeat of a nest: using imagers as biological sensors. *In submission to ACM Transactions on Computational Logic*, April 2008.
- [2] O. Gnawali, B. Greenstein, K.-Y. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, and E. Kohler. The TENET Architecture for Tiered Sensor Networks. *In Proc. 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys’06)*, Boulder, Colorado, Nov. 2006.
- [3] J. Paek and R. Govindan. RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks. *In Proc. 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys’07)*, Sydney, Australia, Nov. 2007.
- [4] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. *In Proc. 3th ACM International Conference on Embedded Networked Sensor Systems (SenSys’05)*, pages 192–204, San Diego, California, Nov. 2005.