WCNN: Wireless Camera Node Network

Alan J. Boguslawski, EE, Andrew D. Flewellen-Gore, EE, and Ping T. Fung, EE

Abstract—**WCNN (Wireless Camera Node Network) is a wireless sensor network composed of environmental sensors and cameras used to monitor wildlife populations. WCNN consists of multiple nodes that capture images when motion is detected and send the images and sensor data to a server. The nodes will also propagate data from other nodes to the server. The server will host the received images and data online for analysis.**

I. Introduction

Wildlife conservation has been a growing concern with many species becoming endangered or extinct. Biodiversity has been decreasing, and there are no signs of it slowing down [1]. Recently, humans have tried to restore wildlife populations by reintroducing species to an area that they used to live in. One example of this was a plan to reintroduce eastern timber wolves in the northeast United States in order to maintain ecological balance and to increase tourism [2].

Wildlife must be monitored to measure how populations change over time, especially when species are being introduced into an ecosystem. Important data collected from monitoring wildlife can be used to track changes in population sizes and how populations migrate to different locations. Data can also include information about the environment, such as weather patterns which might affect the wildlife population sizes and behavior. For example, if timber wolves were to be reintroduced to the northeast United States, it would be important to monitor the newly reintroduced wolves to ensure that they are able to survive. Existing wildlife in the area may also need to be monitored to ensure that the wolves do not negatively affect their populations.

Traditionally, wildlife has been monitored by humans directly observing their habitats. This has the disadvantage that the humans may disturb the wildlife they are observing, which can affect the results. It also requires that people are hired to study the wildlife, which can be expensive and time consuming.

Recent advancements in technology have allowed wildlife to be monitored remotely. An existing technique is to use wildlife cameras that store images on memory cards. A disadvantage to this technique is that the memory cards must be regularly retrieved in order to collect images. This can be time consuming, and it can interfere with the wildlife that it is supposed to monitor. An existing large-scale implementation of this is the Wildlife Picture Index (WPI), a network of wildlife cameras deployed in 14 countries [3]. Images captured by the cameras are used to calculate trends in population sizes over time.

Some wildlife cameras such as the Covert Code Black Scouting Camera contain a cellular modem that can send images as text messages or email [4]. This allows images to be collected without interfering with wildlife. These wildlife cameras are more expensive than cameras without a cellular modem and the operator must pay for cellular service in order to use it. Locations that are in need of wildlife monitoring are often remote and do not have cellular service, so these cameras will not work.

Another solution for monitoring wildlife is TigerCENSE, a prototype wireless sensor network camera designed by students in Dhirubhai Ambani Institute of Information and Communication Technology in Gandhinagar, India [5]. It was designed to monitor populations of tigers with a network of cameras. Initially, it was only capable of single-hop communication, but other students have made improvements to the design that enabled multi-hop communication [6].

WCNN will also be a wireless sensor network with cameras, but each node will also contain humidity and temperature sensors to monitor the environment. The nodes will be designed to be able to be attached to trees and capture images of medium and large mammals during the daytime. A server will collect images and sensor data from the nodes and host them on a website for analysis. The server will also be responsible for controlling the rest of the network.

TABLE I
Specifications

Specification	Value
Image Size	320 x 240 (QVGA)
Node Range	>400 meters
Battery Life	>2 weeks
Maximum Nodes	≥32 nodes
Maximum Hops to Server	≥ 8 hops
Cost Per Node	<$100
Enclosure	Weatherproof
Motion Sensor Range	>10 meters
False Positive Detections	<10%
Animals Detected	Medium-Large Mammals
Light Required for Cameras	Daytime Sunlight in a Forest

A. J. Boguslawski from Bernardston, Ma (e-mail: abogusla@umass.edu).
A. D. Flewellen-Gore from Chelmsford, Ma (e-mail: aflewell@umas.edu).
P. T. Fung from Quincy, Ma (e-mail: pfung@umass.edu).

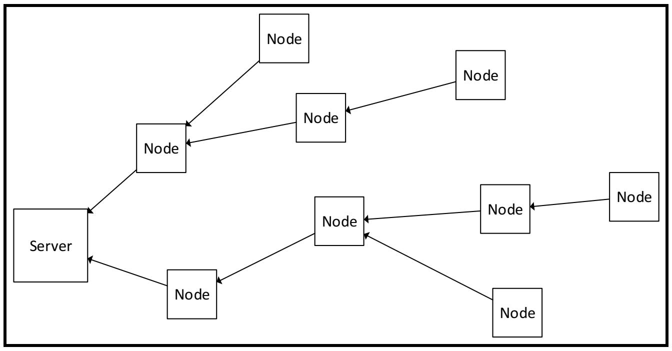

Fig. 1. Network Topology

Some of the challenges associated with such a network include the large size of images compared to the small amount of data used for most wireless sensor networks. A node cannot receive data from multiple nodes at the same time, so the network must be designed to avoid collisions.

It is also important to design the nodes so that they do not interfere with wildlife so that the system gathers accurate information about the wildlife being observed. Animals should not be able to notice the camera nodes, and the nodes should be designed so that they do not release toxic chemicals into the environment. Battery life is also a concern because changing the batteries frequently can interfere with the wildlife, so each node must be designed for low power consumption. The nodes should also be weather resistant because they must be able to survive in a wide range of temperatures and rain. A summary of specifications is shown in table 1.

II. Design

There are two main subunits of our design that will make up our network.

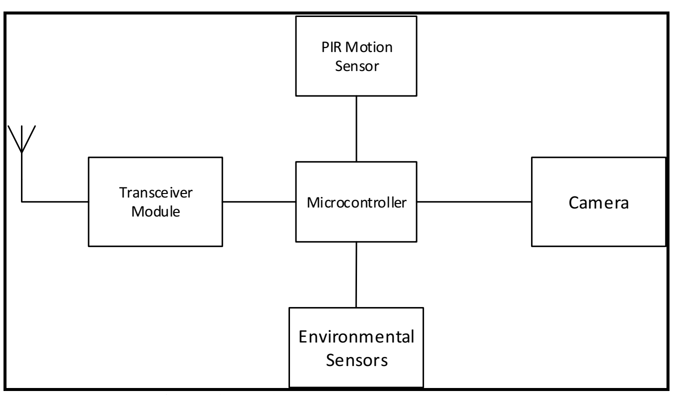

Fig. 2. Camera node configuration

The camera nodes are the building blocks for the wireless network. A passive infrared (PIR) sensor will act as a trigger for the camera. When a warm-bodied animal (or warm object) moves, the PIR sensor will trigger a photo to be taken. For this to occur, the microcontroller will send commands to the camera to obtain JPEG picture data when triggered. Once a picture is obtained and stored on the RAM, the microcontroller will then use the transceiver module to send the image to the server or to other nodes that are closer to the server. The individual nodes will wirelessly communicate in order to move picture data through the network towards the central server. Other useful environmental data, such as temperature and humidity metrics will be sent in a similar fashion. To make sure that data is transmitted towards the server, not away from it, we will first hard code which nodes can communicate with each other and in what direction. As the network functions become more complete we hope to give the nodes an ability to dynamically locate each other, so that the server can determine which other nodes are closer (fewer neighbors to the server). The server will then be able to enforce communication patterns on the nodes, which will discussed in more detail later in this paper.

Fig. 3. Server configuration

The server will communicate with nearby nodes to obtain picture data and environmental data that has traveled through the network. The data will then be uploaded to a website using an internet connection. The Single-board Computer will most likely be an Intel Atom board (pending standing in the Cornell Cup) running Linux. The server will organize and prioritize the order of transmissions that occur in the network. The server will also obtain and store useful information on the nodes in the network. For example, the server will determine which nodes are within range of each other (neighbors) and which nodes have pictures stored in their RAM through periodic polling of the network. Each node will have a randomized address to begin with, but upon initial contact with the server, the server will make sure that there are no address conflictions on the network.

A. Microcontroller

The microcontroller used is a PIC32MX170F256B [7]. It has 64kB of RAM, enough to store a JPEG image. In TigerCENSE, the microcontroller did not have enough room to store images, so images were temporarily stored on SD cards. With this microcontroller, we do not need external memory. The chip is offered in easy to solder SOIC and DIP packages, has a low power mode, supports SPI for communication with the transceiver, and is cheap (less than $5 each). The microcontroller has a maximum clock frequency of 50 MHz but we will operate it at 4 MHz for lower power consumption.

The programs for the microcontroller are written in C using the MPLAB X IDE. Although we have experience programming microcontrollers, this is the first time that any of us have programmed for a PIC32 microcontroller.

B. Camera

We are using a PTC06 camera module. The module contains a VC0706 DSP that performs automatic contrast, brightness, and white-balance functions [8]. It also compresses images to the JPEG format which will make the images small enough to fit on the microcontroller's RAM and reduce the amount of data that will need to be transmitted. The camera can output QVGA-sized pictures; this is the size we plan to transmit over the network.

The camera is controlled by the microcontroller, and has a serial UART communication protocol. We connected the camera to the microcontroller using the PIC32's hardware UART running at 38400 baud. In our final design, we will probably use a different camera that supports a faster protocol; this way, we can maximize the amount of time the microcontroller is in sleep mode for power savings.

C. Transceiver/Antenna

The transceiver modules used for the nodes are HopeRF RFM23BP modules [9]. These modules use the 915MHz ISM band and have a data rate of up to 256 kbps. The module can transmit up to 30dBm, and it supports GFSK, FSK, and OOK for modulation types. We will use GFSK modulation because it has a lower bandwidth than FSK and it is less susceptible to noise than OOK.

The transceiver module uses an SPI bus to communicate with a microcontroller. The microcontroller communicates with the module by reading and writing to 128 registers that are used to change settings, transmit data, and receive data. We used the microcontroller's hardware SPI to communicate with the microcontroller, and we used an additional GPIO pin connected to the transceiver's interrupt pin that tells the microcontroller when a packet is received.

To determine the range of the transceivers, we conducted a test. One node was programmed to transmit packets every 0.25 seconds containing a single data byte alternating between 0x00 and 0x01. In addition to the data byte, the packets also contained a preamble, sync word, header, packet length, and a CRC. Another node was programmed to receive the data and blink an LED. It would turn on the LED when it received 0x01 and turn off the LED when it received 0x00. For the test, we walked with the receiving node until the LED stopped blinking at a steady rate and measured the distance between the transmitter and receiver with an Android smartphone using "GPS Tape," an app that uses GPS to measure distance between opjects. The test was repeated with different data rates. The test was conducted in an area with many buildings that seemed to have a significant effect on the range. When we walked behind a building, the LED stopped blinking at a shorter distance than when we did not go behind buildings. This is expected because reinforced cement walls and brick walls attenuate 915MHz signals more than free space [10]. In an open area, the range may be farther. Quarter-wavelength monopole wire antennas were used for the test. The results are shown in Figure 4.

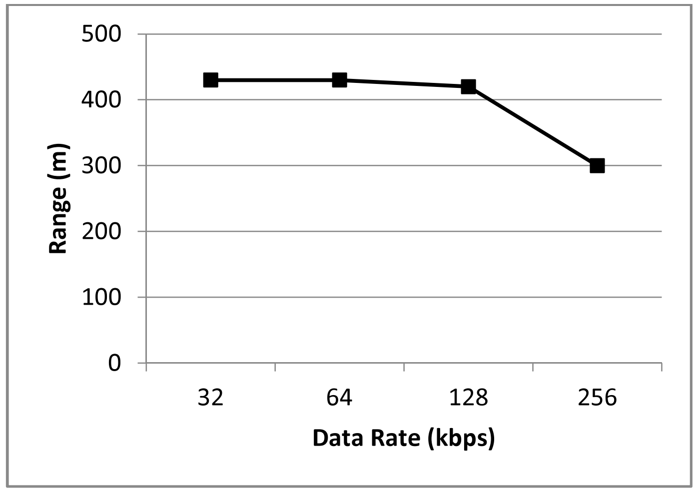

Fig. 4. Graph of range with different data rates

Wireless communication involves a signal that propagates from the transmitting antenna through the air towards the receiving antenna. Only a small fraction of the signal that gets transmitted makes it to the receiver. One way to maximize the power of the signal at the receiver is to increase the transmission power, but the FCC has a 30dBm power output limit for 915MHz transmitters [11].

The signal can also be focused in the direction of the intended target using a directive antenna. For now we are only using a simple monopole antenna, but we can increase the communication performance by using more directive antennas. A loop antenna can offer more directivity than a monopole antenna [12]. However, we don't want the antenna to be too directive, as that may limit the capability of the node network system. Each node has to talk to multiple nodes that can be in any direction; therefore, directing the signal in one specific direction is not useful. However, we can try to direct the entire signal onto the ground plane. Therefore the spherically propagating signal can now be changed into a circularly propagating signal.

D. Sensors: Temperature, Humidity, and Passive Infrared

The three sensors on each node are for capturing the temperature, humidity, and infrared signal. The passive infrared sensor (PIR) generates a pulse when it detects any movement [13]. A PIR sensor was selected over types of motion detecting sensors such as ultrasonic sensors because it only detects motion of objects that emit infrared radiation, so there will not be false positives if tree branches move from the wind. PIR sensors also use less power than ultrasonic sensors. We are using a Parallax PIR sensor. The PIR sensor's range can be adjusted to suit the application by changing a jumper. If the PIR sensor's operational radius is small, then setting the PIR sensor to a lower range may be used to shield away unnecessary infrared noise in the background. Currently, the PIR sensor has a wide sensing radius, so the edges of the sensor will need to be blocked in the final version so that the camera does not take pictures when an animal is out of the camera's view.

The temperature and humidity sensors are located on the same component, a MaxDetect RHT03 [14]. The RHT03 uses a proprietary single wire serial interface based on PWM. Temperature and humidity data is sent in packets of 40 bits. Short pulses ranging from 20-40 microseconds represent a 0 and long pulses of up to 80 microseconds represent a 1. These pulses are read by the microcontroller which uses a timer to determine the length of each pulse.

E. *Power Supply*

All components in the nodes use 3.3 volts except for the transceiver modules. The transceiver modules cannot transmit at their maximum power when powered by 3.3 volts, so we will connect the transceiver modules directly to battery power. The transceiver modules are capable of measuring the battery voltage, so another advantage of connecting them directly to battery power is that we can monitor the remaining battery life.

The type of batteries we will use is to be determined, but we will likely use either three 1.5 volt alkaline batteries, which will output 4.5 volts for the transceiver module or a 4 volt lead acid battery.

To get 3.3 volts to power the rest of the circuit, we will use a Pololu 3.3v 500mA Step-Down Voltage Regulator [15]. This regulator module contains a buck converter circuit, which is more efficient than a linear voltage regulator. The efficiency of this regulator is above 90% when the input voltage is 5 volts [16].

F. Power analysis

The nodes must be designed for low power so that the battery life meets the specifications. Most of the time, the microcontroller and other components will be in a low power mode. After a predetermined interval such as every 30 seconds, all nodes will wake up to transmit and send data. During this time, the nodes will send information back to the server indicating if they have a new image. The server will then determine the order for the nodes to send images to the server. A node will also wake up briefly when motion is detected, which will cause a picture to be taken and stored.

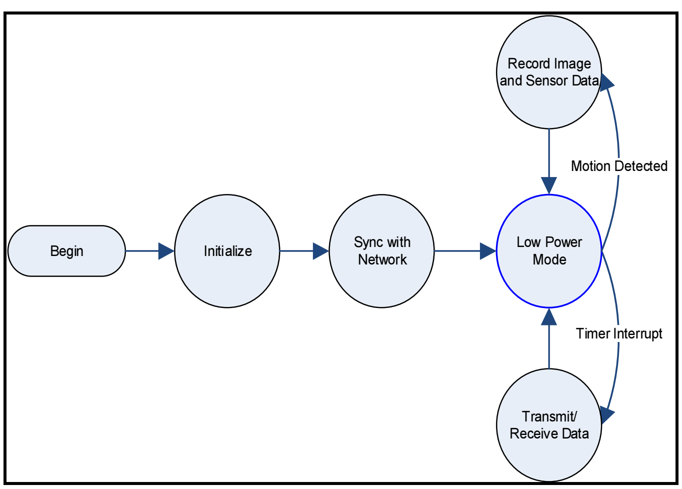

Fig. 5. Program flow

We estimated the power consumption assuming the nodes will be in a low power mode for 29 seconds, transmitting for 0.5 seconds, and receiving for 0.5 seconds. We did not include the power consumption of taking a picture because pictures are expected to be taken infrequently and have a small effect on the total power consumption. The power consumption of each stage was calculated by summing the power consumption of each component and multiplying the total by the fraction of time the node will be in that stage. For these calculations, we assumed that the power supply for the components using 3.3 volts had an efficiency of 95%. The estimated average power consumption is 43.82mW. Therefore, if the nodes are powered by three 18000mAh Energizer alkaline D batteries [17] (totaling 81Wh), they will last approximately 77 days.

TABLE II
POWER CONSUMPTION ESTIMATION

	Low Power	Transmit	Receive
Microcontroller (4 MHz clock) (3.3v)	145.2 µW	6.6 mW	6.6 mW
Tranceiver (4.5v)	4.5 µW	2.475 W	112.5 mW
PIR Sensor (3.3v)	165 µW	165 µW	165 µW
Temp/Humidity Sensor (3.3v)	165 µW	165 µW	165 µW
Power Supply	25 µW	365 µW	365 µW
Total:	504.7 µW	2.482 W	119.8 mW
Estimated time in stage per 30 seconds	29 sec	0.5 sec	0.5 sec
Average power consumed per period	487.88 µW	41.37 mW	2.00 mW
Average Power:	43.86 mW		

G. *Cost Estimation*

We estimated the cost of a node based on ordering parts for a quantity of 100 or more. The estimated cost was $61.45, but the cost might become lower as we plan on selecting a cheaper PIR sensor and camera. The estimation does not include the cost for passive components, PCB, and enclosure.

TABLE III
COST ESTIMATION

Component	Unit Price	Other info
PIC32MX170F256B	$3.10	>100 quantity from Mouser
HopeRF RFM23BP	$8.80	From Anarduino
Mini TTL Serial Camera	$28.76	>100 quiantity from Adafruit
Parallax PIR Sensor	$9.34	>5 quantity from Mouser
MaxDetect RHT03	$7.96	>100 quantity from Sparkfun
Pololu 3.3v 500mA Step-Down Voltage Regulator	$3.49	>100 quantity from Pololu
Total (per node)	$61.45	

III. Project Management

TABLE IV
MDR Goal Status

Proposed Deliverable	Status
Demonstration of communication between transceiver modules connected to microcontroller and server.	Communication works between nodes. We do not yet have a server, so we cannot test communication with it yet.
Demonstration of capturing and storing images with camera.	We are able to capture images with a microcontroller.
Demonstration of reading data from PIR and environmental sensors	We are able to detect motion with the PIR sensor and read data from the temperature/humidity sensor.

We met all of our goals for the MDR. The next steps that we have to do are to set up the server to host a webpage and communicate with nodes. We will need to program the nodes so that they know how to act in a network with other nodes. We will also need to design PCBs and enclosures for the nodes and the server.

We divided the work so that we can work independently, and then combine our work. For the MDR, Alan worked on the transceiver modules, Andrew worked on the camera, and Ping worked on the sensors. Next, Alan will work on the server, Andrew will work on PCB design, and Ping will work on networking.

Our team has weekly meetings with our advisor. We also meet on our own to discuss the project and to help each other with their parts. Alan and Andrew's expertise are in circuit design and embedded programming. Ping's expertise is in networks and electromagnetics.

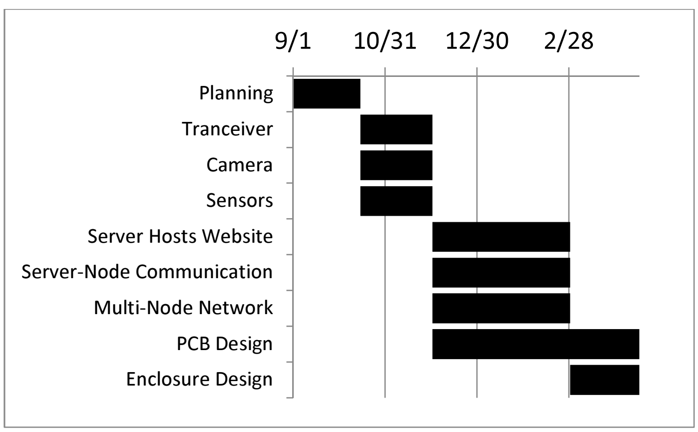

Fig. 6 Gantt chart

IV. Conclusion

For the MDR, we delivered individual working wireless nodes with point-to-point communication. We were able to read sensor data, capture images, and transmit data between nodes.

The next part of our project will involve networking multiple nodes together. To avoid collisions and other inefficiencies, protocols must be introduced. Our current networking design involves putting the nodes in different rings around the central server. The nodes that are part of the ring closest to the central server will experience the most traffic, so we will put considerable attention to that. We will also adjust the rate that the pictures are taken to ensure that the network won't be too congested.

An anticipated challenge will be trying to get the transceivers to communicate with each other in a forest. We have conducted multiple tests for the transceivers and the data shows that the operational range of the transceivers in a line-of-site can easily reach 400 meters. However, once we place these devices in the forest, many unexpected results may occur. There will likely be trees and other plants between two separate nodes, which may attenuate or block the signals. The signal polarization will also be affected as it impinges on surfaces, which further reduces the signal strength.

We will also need to host a website with the server that displays images and sensor data. The server will also need to communicate with nodes. We will also need to design PCBs and eventually create enclosures.

Acknowledgment

We would like to thank Professor Paul Siqueira for those endless hours he spent to provide us with invaluable help throughout our designing process. It was and still is a pleasure for us to work with you!

We would also like to thank Professor Krishna and Professor Pishro-Nik for the constructive criticism that they provided.

References

[1] S. H. Butchart *et al.*, "Global Biodiversity: Indicators of Recent Declines," Science, vol 328, no 5982, p.1164 – 1168.

[2] A. Hodgson, "Wolf Restoration in the Adirondacks? The Questions of Local Residents," Wildlife Conservation Society, Working Paper No. 8, 1997.

[3] "Wildlife Picture Index," [Online]. Available: http://wpi.teamnetwork.org. [Accessed Dec. 2, 2014].

[4] "Special Ops Code Black," [Online]. Available: http://covertscoutingcameras.com/products/cameras/special-ops-code-black/. [Accessed Dec. 12, 2014].

[5] R. Bagree *et al.*, "TigerCENSE: Wireless Image Sensor Network to Monitor Tiger Movement," in *Proceedings of the 4th international conference on Real-world wireless sensor networks*, Berlin, Heidelberg, 2010, pp. 13-24.

[6] A. Varshney *et al.* "Improved Wireless Image Sensor Network to Monitor Tiger Movement," in *Proceedings of the 7th International Conference on Wireless Communication and Sensor Networks*. 2011.

[7] Microchip, "PIC32MX1XX/2XX 32-bit Microcontrollers with Audio and Graphics Interfaces, USB, and Advanced Analog," datasheet, 2011 [Revised 2014].

[8] Guangzhou Putal Communication Technology Co. Ltd, "PTC06 串口摄像头模块 使用说明书," datasheet, 2012.

[9] HopeRF Electronic, "RFM23BP ISM Transceiver Module," datasheet, 2011.

[10] O. Felekoglu, "Propagation and Performance Analysis for a 915 MHz Wireless IR Image Transfer System," M. S. thesis, Naval Postgraduate School, Monteray, CA, 2005.

[11] FCC 47 CFR 15.247.b.1, Oct. 1, 2013.
[12] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Hoboken, NJ: Wiley, 2005.
[13] Parallax Inc., "PIR Sensor (#555-28027)," datasheet, 2012.
[14] MaxDetect, "Digital Relative Humidity & Temperature Sensor RHT03," datasheet.
[15] "Pololu 3.3V, 500mA Step-Down Voltage Regulator D24V5F3," [Online]. Available: http://www.pololu.com/product/2842. [Accessed Dec. 12, 2014].
[16] Intersil, "ISL85415 Wide Vin 500mA Synchronous Buck Regulator," datasheet, 2013.
[17] Energizer, "Energizer E95," datasheet.