SEA-LABS

+ Students: Joann Chou with Matthew Bromage, Christopher Chan, Michael Maroney generator; Lionelle Wells
+ Faculty: Katia Obraczka; Stephen Petersen; Donald Potts
Department of Computer Engineering, University of California, Santa Cruz

Summary

SEA-LABS is a low cost, power-efficient wireless sensor network for sustained, real-time monitoring of shallow water coral reefs. Each sensor node consists of three primary elements: a "POD" that houses most of the electronics (including a microcontroller, memory, and battery) designed to operate submerged up to 20 meters below sea level, sensors (temperature, light, pressure, and salinity) attached directly to the reef, and a floating buoy containing the communication subsystem. Sensor nodes form an ad hoc network that relays sensed data to a sink, which uploads readings to a database accessible over the Internet. The first phase deployment at Midway Atoll, Northwestern Hawaiian Islands was in August 2005.

Mothership Functionality

The Base Station computer, affectionately called the "Mothership", holds all software that will handle receiving packets, parsing the packets, and uploading the data to an online database. In order to make sure the packets are continuously being received as they are sent, the computer runs a cron job that checks every minute to see if the SEA-LABS program is still running.

The Packet

To signify the beginning and end of every packet, the packets are framed with 0xFF. Two of the packet types are:

+ The Data Packet for Temperature, Visible Light, and Pressure (Once every 15 minutes).
+ The Data Packet for Pressure (Every second for 5 minutes, 4 times a day)

Everything in each type of packet is the same except for the payload. Now, what is exactly inside of each packet? Within each packet is a Source ID, Destination ID, Packet type, Nonce, Payload, and CRC. Once the packet is received, the payload is ready to be parsed.

The Data

Once the packets are received, we must analyze what is inside the payload of each packet. The packet type will tell us if the data packet is a Temperature, Visible Light, and Pressure packet or a Pressure packet. Once we have the payload, we can convert the voltage readings for each sensor into readable data.
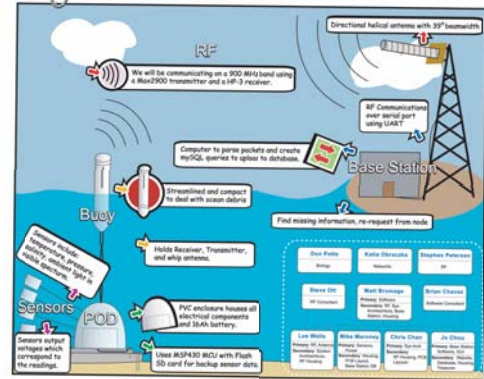
JDBC API

In order to have all the information readily available, the data must be uploaded to the online database. Using the JDBC API, we connect to the database using our username and password. With the help of the executeUpdate method from the Statement class, the data automatically gets inserted into the database. The database contains four tables—one for each sensor. The structures of the tables in the database are as such:

```
'sid' bigint(20) NOT NULL,
'sampledate' date default NULL,
'sampletime' time default NULL,
'sensor' float default NULL,
```

PHP GUI

To view all of the information, we must have all the data be viewable. Simply so, the ease of PHP will handle what the user can see. The PHP page on the website will show all the readings as they are sent to the base station computer, dubbed "Mothership".

CARNIVORES

+ Students: Joann Chou with Graduate Student Vladi Petkov
+ Faculty: Katia Obraczka; Cyrus Bazeghi

Summary

CARNIVORES is a small device which attaches to a dog collar and takes in accelerometer readings. The readings are taken along three axes and analysis is done using a square wave. The device is designed to display the movements of any carnivorous animal that has the collar around its neck.

The Packet

The packet for the carnivores also signifies the beginning and end of a packet by framing the packet with 0xFFFF. There are 6 fields inside each packet that store information for each accelerometer reading—2 fields per axis:

```
       [X+, Xtotal, Y+, Ytotal, Z+, Ztotal]
offset   0    1     2    3     4    5
```


Receiving Data

Still in the beginning phases, mock data is necessary for the packet to look realistic. Two ideas were explored:

+ Send data by connecting two computers via serial cable.
+ Create a mock connection by creating a fake socket, then send the test packet through the connection.

Future

+ Have the data show up as a real-time graph using Java. (Similar to an oscilloscope.)

Illustrations and Design by Jo Chou