Distributed Sensing and Data Collection Platform Development

(Case Study: Solar Energy Viability Study)

CSC 775 - Final Report

Robinson Udechukwu*
Amlakawit Medhin†

December 1, 2013

*rnudechu@ncsu.edu
†aamedhin@ncsu.edu
## Contents

1 Introduction ........................................... 1

2 Related Work ........................................ 2

3 Design Model ...........................................
   3.1 Payload Header .................................... 2
   3.2 Server Design ..................................... 2
   3.3 Client Design ..................................... 5
      3.3.1 Sensor Reading ................................ 7
   3.4 User Design ....................................... 7
   3.5 Developer Code ................................... 7

4 Experimental results and Analysis ........................ 8

5 Conclusion ............................................. 10

6 Appendices ............................................. 10
Abstract

In this project, we plan to introduce a distributed sensing and data collection platform for wireless mesh networks. The platform has two main components: the Sensor Platform (Client) and the Mesh Platform (Server). The Sensor Platform program takes sensor readings from sensing devices, buffers the readings and sends it to a server. In addition, this program will self-configure itself on the receipt of configuration command from a server. The Mesh Platform program is distributed program that receives data from the sensor nodes, stores the received data in a centralized storage, and manages sensor nodes.

Each of the sensor nodes are composed of sensing devices interfaced with a BeagleBone Black. The client side software of the platform will be running on the BeagleBone Black. The server end of the system is a wireless mesh network of Linux machines. The distributed server software will be running on each one of the mesh nodes. Sensor node - Mesh node communication is via single-hop WiFi broadcast/multicast mode.

Design and development goals of the designed system are not tied with any kind of specific sensing application, it is generic. Therefore, the system is applicable to a wide variety of sensing and data collection applications. To validate the value of this project we will use the platform to study solar energy viability and present results.

1 Introduction

In a move to cut cost of energy expenditures, companies are investing more time in investigating in how renewable source of energy, such as sunlight, wind, hydropower, and geothermal can be used to power their industries. During these investigations, companies must wage whether the selected renewable energy selected are sustainable and reliable for their facility location. The cost of installing equipment to test the viability of these solutions can discourage interest. We propose building a practical solar testing device that can assist in determining the viability of a location at a fraction of the cost of installing solar panel.

To achieve this, we developed a sensor platform based on [1] designs on a BeagleBone Black embedded processing system. The design provides a flexible architecture for any sensor reading devices such as microphone or a wind speed detector, to be plugged into it and monitored. To demonstrate the usefulness of this system, we are using a solar sensor to detect light intensity to determine whether a location is viable to provide solar energy.

We implemented this project in the span of two phases:

- Phase 1: Develop single server platform. Figure 1 illustrates this and Figure 7 shows the flow.
• Phase 2: Introduce distributed messages into the server platform, extending it to propagate SET instructions to clients outside a single server’s reach. Figure 2, 3, 4, and 5 illustrates this and Figure 8 shows the flow.

![Bird's Eye View Distributed Server Topology](image_url)

Figure 2: Distributed Server Case: User sends USER_SET message to a server. This message contains a set of instructions for the clients to perform.

2 Related Work

Sensor networks are a common tool used by the military for monitoring environmental and physical conditions such as temperature, intrusion detection, and sound detection. Sensor networks can aid disasters relief situations by detecting and alerting the public and emergency facilities about a disaster minutes/hours before it happens. Sensor networks can be stationary or mobile via being attached on military personnel/units or an unmanned aerial vehicle (UAV). Sensor network often rely on a mesh network for communicating, using a distributed protocol for communicating between each others and the outside.

3 Design Model

The Sensor platform can be viewed as three agents: the server, client, and the user. Each agent share sa common payload structure but have a different topology of components which it uses to digest messages received and sent. The following subsections will describe their flow, the message payload header structure, and each agent’s components and their uses.

3.1 Payload Header

Each payload structure shares a common payload structure’s header. This is illustrated in Figure 16. Each message has an action code which describes the message purpose, this can be seen in Figure 17.

3.2 Server Design

Server maintains several databases and managers to handle the messages it receives. This is illustrated in Figure 9. Only the server who receives the USER_SET message from the user sends a SET message to other servers. All servers broadcast SET messages to their clients.

• Device Database
  – Maintains a database of devices the server has received Heartbeats from
Figure 3: Distributed Server Case: The receiving Server turns the USER_SET to a SET message and unicasts it to each mesh node within its reach.

Figure 4: Distributed Server Case: Each mesh node broadcasts a copy of the SET message to its clients address.

Figure 5: Distributed Server Case: Clients respond back with either a SET_ACK or SET_NACK depending on whether the client can operate on the received SET.
Figure 6: Client Program Flowchart

Figure 7: Single Server Program Flowchart
3.3 Client Design

Client maintains several managers to handle the messages it receives. This has been illustrated in Figure 10.

- **Client Capability Manager**
  - Maintains each capabilities a client may have (solar, temperature, etc.)
  - Client Capability contains the name, max frequency, and representation for a capability
  - Capabilities are set in the config.properties file

- **Client Measurement Manager**
3 DESIGN MODEL

- Maintains each measurements a client has taken
- Client Measurement contains the name, value, representation, and time stamp of sensor reading

- Sequence Number Manager
- Maintains each sequence numbers of messages the client is expecting back from the server
- Sequence Number contains its integer (sequence number), timestamp (creation time), expiration time
3.3.1 Sensor Reading

The solar board is rated 4.5V open voltage and 100mA short circuit. However the analog inputs of the Beaglebone Black supports a maximum of 1.8V. We used 3 Kohm resistors to divide the voltage to ensure the input voltage to the AI pins of Beaglebone Black doesn’t exceed the limit. The 47ohm resistor is to reduce the current that flows to Beaglebone Black pins.

![Solar Cell Wiring](image)

Figure 11: Solar Cell Wiring

3.4 User Design

User maintains a manager to handle the SET operation. This has been illustrated in Figure 12.

- **User Set Buffer**
  - Maintains a database of all the Set Operations it has stored
  - Set Operation contains an instruction set for clients. The receiving server distributes the SET operation among its fellow mesh nodes.
  - Set Operation contains the name, operation, frequency, creation time, expiration time, and their sequence number

![Logical Topology: User](image)

Figure 12: Logical Topology: User

3.5 Developer Code

Developer code was developed for testing and easing the retrieval and accessing the reports over time.

- **Client Test**: Stub code used to simulate the different conversations for the Sensor Platform.
- **Email Developer**: Used to automate the sensor reports sent to the developers.
- **Parse Logs**: Used to parse the sensor reports for creating Matlab arrays
4 EXPERIMENTAL RESULTS AND ANALYSIS

Data collection took place uninterrupted over a four day interval at Wolf Ridge Apartment Complex at Centennial Campus. The solar cell was attached to a screen door facing 273 degrees West with no direct obstructions in its way. Figure 18 illustrates the solar cell’s view. The experiment involves a single server with a single BeagleBone Black client using a solar cell for sensor reading. The solar cell used is rated for 4.5V open voltage and 100 mA short circuit. The BeagleBone Black client were set up to send an heartbeat every 20 seconds, perform a measurement every 3 minutes, and report its measurement every minute and half. We realized after the experiment began that we were inefficiently running reporting timer twice for every measurement taken. Fortunately this inefficiency has no weight on the data received and just results in wasted CPU clocks for the client. Each client time were setup with a network time protocol (NTP) server and the correct timezone, for synchronicity.

From Figure 13, the solar cell observed peak readings around the same time ranges except for November 27. November 27 was also the only date to experience a slight spike in its readings during peak hours. Based on weather reports, Raleigh experienced a cloudy day with light rain on November 27. This forecast would explain the hour delay in the start time of peak readings and the cloud coverage would explain the spikes in readings. During the span of this experiment, Wolf Ridge generated 173 - 179 milliwatts of power during peak hours. Table 1 shows the observed peak readings time ranges and the their values.

A parallel data collection took place over a two day interval at Clairmount at Farmgate with the same equipment setup as Wolf Ridge Apartment. The solar cell was placed next to a window facing West. Table 2 shows the observed peak readings observed in Figure 14 during this experimentation. In comparison with Wolf Ridge readings, peak readings occur later in the day with smaller average readings and power generated. Figure 15 shows an graph with the two readings overlaid on the same graph. Based on these drastic difference, we believe Clairmount at Farmgate’s solar cell placement within the home and human interaction within the home (such as turning on the lights) affected the readings gathered. Based on this assumption, we can stipulate that lights were routinely shut off in the solar cell’s room after 10:30 PM during the observed period.

Power Generated Code Example:

\[
\text{Measured voltage, } V = 5\text{V} \\
R = 47\text{ohm (the value of the resister connected in series)} \\
\text{Voltage on this resistor} = 5/3 = 1.67 \\
\text{Power on the resister} = (V*V)/I \\
(1.67*1.67)/47 = 59.33\text{mW} \\
\text{Total Power} = 3*59.33 = 177.99\text{mW}
\]
4 EXPERIMENTAL RESULTS AND ANALYSIS

Figure 13: Four Day Analysis: Wolf Ridge Apartments

Figure 14: Two Day Analysis: Clairmount at Farmgate

Figure 15: Wolf Ridge vs. Clairmount at Farmgate
5 Conclusion

Our project’s mission was to build a flexible sensor platform for a distributed system. To achieve this, we split the platform design into two phases: Single Server and Distributed Server. Single server implementation would receive messages from a collection of BeagleBone Blacks Clients and maintain their readings in a self-contained database. The Distributed Server built on top of the Single Server implementation allowing servers to forward SET operations between themselves and to their clients if they were the originating server to receive the SET messages from a User.

To validate the value of this platform, we performed two parallel experiments to determine the viability of a location for solar energy. Based on the analysis, Wolf Ridge Apartment at Centennial Campus came out to be a more viable location for solar energy than Clairmount at Farmgate. Using this platform, we were also able to determine when an obstruction had occurred based on historical data.

6 Appendices

![Figure 16: Payload Structure’s header](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>String</td>
</tr>
<tr>
<td>Action Code</td>
<td>Integer</td>
</tr>
<tr>
<td>ID</td>
<td>String (Hardware’s MAC address)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Action Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HEARTBEAT</td>
<td>Message from the Client, informing the server of its presence and capability</td>
</tr>
<tr>
<td>1</td>
<td>SET</td>
<td>Message from SERVER, for Clients to send SET commands</td>
</tr>
<tr>
<td>2</td>
<td>REPORT_SENSED_DATA</td>
<td>Message from the Client, reporting sensed data</td>
</tr>
<tr>
<td>3</td>
<td>ACKNOWLEDGE_SENSED_DATA</td>
<td>Message from Server, acknowledging readings received</td>
</tr>
<tr>
<td>4</td>
<td>SET_ACK</td>
<td>Message Acknowledging a SET message</td>
</tr>
<tr>
<td>5</td>
<td>SET_NACK</td>
<td>Message from the Client about requested reading which it can not deliver</td>
</tr>
<tr>
<td>6</td>
<td>USER_SET</td>
<td>Message from USER with the SET values to send out</td>
</tr>
<tr>
<td>7</td>
<td>SERVER_BROADCAST_CLIENT_SET</td>
<td>Message to tell the server to broadcast set to clients (internal control)</td>
</tr>
<tr>
<td>8</td>
<td>SERVER_BROADCAST_MESH_SET</td>
<td>Message to tell the server to broadcast set to mesh nodes (internal control)</td>
</tr>
<tr>
<td>9</td>
<td>CLIENT_BROADCAST_HEARTBEAT</td>
<td>Message to tell the client to broadcast heartbeat (internal control)</td>
</tr>
<tr>
<td>10</td>
<td>CLIENT_BROADCAST_MEASUREMENT</td>
<td>Message to tell the client to broadcast measurement (internal control)</td>
</tr>
</tbody>
</table>

![Figure 17: Message Action Code](image)
Figure 18: Wolf Ridge Solar Cell View
References