High-level design

Our system begins with the sensor node, which polls a sensor and passes the information to a base station, which in turn forwards the packet to the processing client through a serial port. The processing client converts the raw data into power usage information, and submits it to a database. A client can view the data stored in the database through a web browser, and the information will be displayed in a graph showing the energy consumption along a timeline. See Figure 1 for a pictorial description.

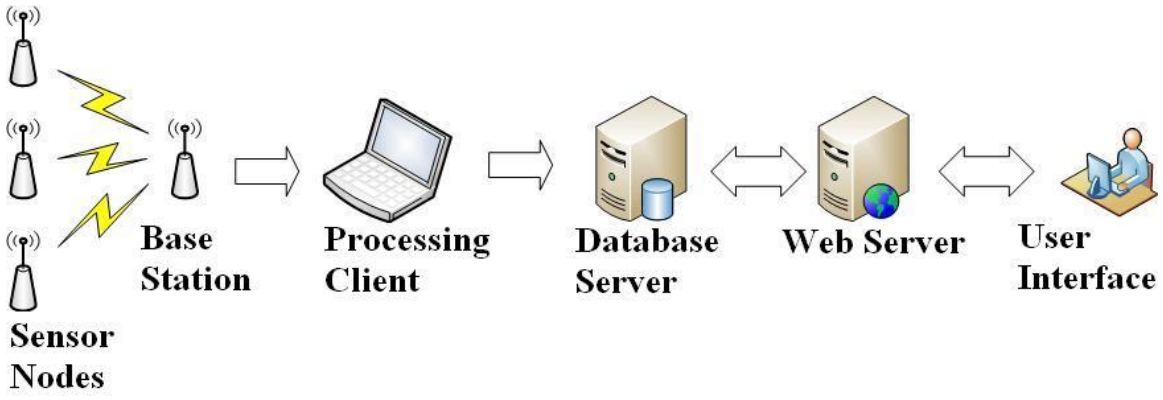

Figure 1: System Architecture

Detail design

Sensor nodes and base station

The sensor node begins by polling the sensor and collecting readings over a period of time, by default five seconds. The sensor node polls sensor data directly from the Analog-to-Digital Converter (ADC). The rate at which this occurs is also configurable, and defaults to 3125 Hz.

The sensor node is also capable of dumping the data it collects to its built-in serial port. A small button on the node, labeled "user," initiates collecting a trace of the current and voltage sensors over a few seconds. After it has filled its internal buffer, the node will then break the buffer into pieces, wrap each piece in a packet, and send it to the serial port, where it can be analyzed. To initiate another trace, the user must press the "user" button again.

After five seconds of data have been collected, the sensor node will send the data to the base station (see Figure 2), using the Collection Tree Protocol (CTP).

In order to balance sampling the sensor and sending the data, the sensor node calculates a running average of all readings since the last send. This setup minimizes network traffic, as well as power usage in the sensor node. If, however, the node is currently capturing a trace because its "user" button was pressed, it will also save each reading in its internal buffer until it is full.

The base station listens for packets from the sensor nodes. When a packet is received it forwards the information over the serial port and returns to listening for more packets.

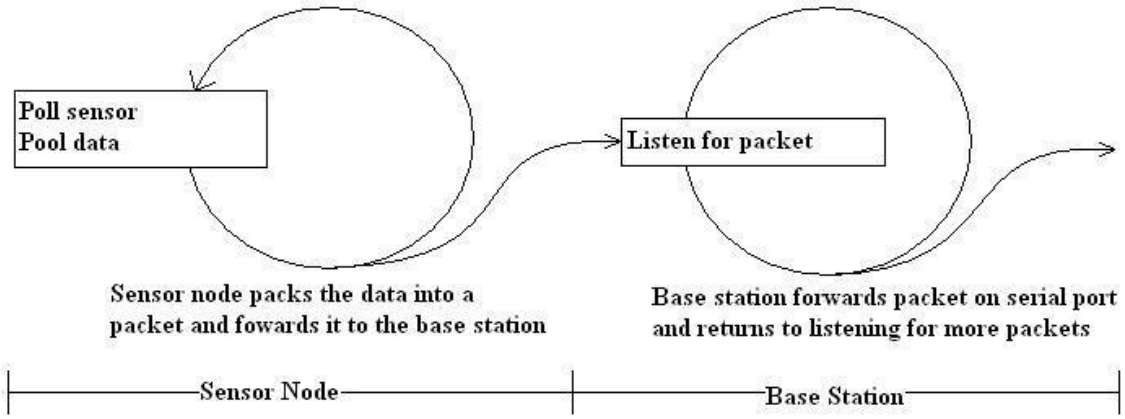

Figure 2: Data from sensor node to base station

Processing client

The processing client is written in Java and is split up into two distinct threads. The first thread listens to the serial port, and when a packet is received it parses the packet and inserts the information it extracts into a queue. After the information has been added to the queue it returns to listening for packets on the serial port (see Figure 3).

The second thread pulls information out of the queue and turns the raw sensor data into power information. Afterwards a connection to the database is established and the converted data is submitted to the database.

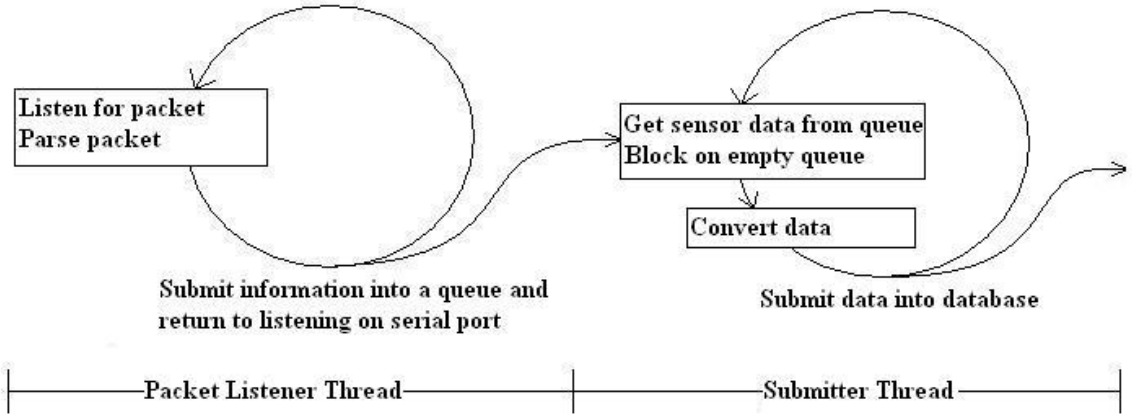

Figure 3: Client processing data

Webpage

The web server allows clients to view their power usage in real time. In this design, the web server is actually the same computer as the database server. We made this decision for the sake of efficiency and compatibility.

The web page itself contains a script that makes requests to the server. The data are returned in Extensible Markup Language (XML) format, and the script can use the XML Document Object

Model (DOM), a standard for representing XML data as a tree of objects, to manipulate the data.

The script actually makes three types of requests to the server. In the first type, the script asks for a list of known groups from the database. In the second type, the script has selected a group, and asks for a list of nodes that belong to the group. In the third type, the script has selected a node in a group, and continually queries every five seconds for power usage data from the database.

The web server responds to requests from the web page script by executing PHP Hypertext Processor (PHP) scripts. These scripts query the database for the desired information using a stored procedure, and emit an XML document that is returned to the web page script.

When the web page script receives power usage data, it feeds the XML document into a Flash-based chart. This chart and the data it displays are controlled by various inputs on the web page. The script is aware of these inputs, and adjusts the requests and the operation of the chart accordingly.

These inputs are as follows. The first set of inputs is located above the graph. It allows the user to select a group and a node inside that group. When the user selects a new group, the script automatically refreshes the list of nodes inside that group. When the user selects a new node, the chart is then reloaded with the data from the new node.

The second set of inputs is located below the graph. The user can use these inputs to specify a span of time containing the data to request from the server. The user can specify a specific date, or any time relative to now.