Query based Software Application for Weather Monitoring System

Pramod Srivatsa Krishna Murthy
Prospective Research Scholar
Mysore, Karnataka state
INDIA.

Prasanna Puttaswamy
Prospective Research Scholar
Mysore, Karnataka state
INDIA.

ABSTRACT

Environmental Disasters form the most important aspect that affects day to day life of human society. The main causes include threats from Floods, Tornado etc… So, weather forecasting forms most surveyed activity. Recently, emerging technologies such as WSN are being used for accurate weather forecasting. This paper presents one such software application bearing in mind some of the hardware that could be used in WSN weather monitoring Implementation.

General Terms

Wireless Networks, Security, Mobile Computing & Software Implementation.

Keywords

WSN, Query based Weather monitoring System.

1. INTRODUCTION

Wireless sensor network (WSN) is an emerging technology which has made significant contributions in making our lives felicitous. Due to features like ease of use, scalability, robustness, mobility and low power consumption it has been deployed in many fields. For example, in environmental monitoring, military applications, security, health monitoring and fleet monitoring etc. In the field of WSN, publish-subscribe (pub/sub) and query-based are the major services provided. The most popular one is the query-based information extraction.

Several applications on query based WSN systems have already been deployed in different applications. The WDSS [1] concept is one such concept which is designed for managing, sharing & accessing weather related information's. Nodes in this system are incorporated with a designed service model for querying & dataset accessing based on query from the agent. Although this system manages to retrieve the datasets from local weather data & global metadata, the system's huge networks makes it more time consuming for the query to be answered.

Another system with a feature where the end user can make the query through any hand held devices like PDA, Mobile phones etc. with an application running in it, where the required info on weather can be queried [2] is directly connected to a weather information server which gets the data's from a website supplying a free weather forecast. The application built on J2ME (Java 2 Platform Micro Edition) in the device enables the user to send query through an interface which connects to information server for further processing of the query to generate required result. The result obtained may not be a novel one as there may be chances that the free weather forecasting website's updating intervals frequency.

This paper presents one such novel concept on Query based weather monitoring system, where there is no need of dependency on any foreign source of medium for result generation as this system is a self-sustaining network. Here, the system is self dependent as it forms a complete network with several sensor nodes each containing different sensors for sensing different parameters which are deployed in required geographical locations & these are monitored by a gateway station with a Base Station to input query. The proposed system accepts user queries through Base station & presents the end user with result after invoking & executing relevant services which helps in prediction of weather forecast.

2. SYSTEM ARCHIETECTURE

Base station, Sensor node & respective sensor's are the major components forming a basic weather system. This system is designed considering three different sensor nodes with each having three different sensors to sense three different parameters i.e. Temperature, Humidity & Wind. The Query is requested by user from the base station which is sent to the server for processing and the returned results are displayed on the base station. In this system, the gateway node is responsible for serving the queries/requests from the clients. Upon receiving the query it parses the query and then validates the node number specified in it. Then it forwards the remaining data items in the query to the sensor node specified. The sensor node is responsible for identifying the type of sensor (temperature, humidity or wind) from which the data is requested. Then it performs the function requested (avg, max or min) over the requested range of values and returns the results to the gateway node. Gateway node then returns this result to the client.

In this system socket connection has been used to connect between base station and server. So this enables the base station on one computer and server on other computer to get connected wirelessly or both the server and base station can be connected to each other (using cal host) via ports. A block diagram of the presented systems is given below.

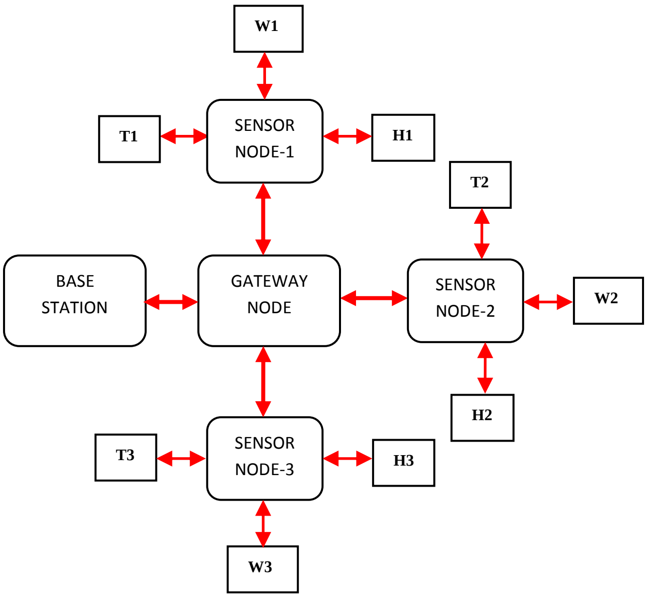

Figure 1 Block Diagram of WSN Weather monitoring system

3. HARDWARE CONSIDERATIONS

The developed system under this project can be utilized on 'Gumstix' (a type of Mote). Gumstix is named as such cause of its size which resembles that of a Glue stick. Gumstix is actually a small computing system which comprises of Memory (RAM & ROM), Processor, OS & Networking capabilities. There are basically two manufacturing companies for Gumstix, one is 'Overo' series under Texas Instrument & the other is 'Verdex Pro' series under Marvell XScale. Upon the two said models 'Overo Earth' is more advanced with Enhanced memory & several special features like microSD slot, I2C, UART, SPI pins included in it which doesn't come with the latter 'Verdex Pro' model. Although these two said models use different processors the speed of them remains same at 600MHz i.e. 'Overo Earth' model uses OMAP 3503 application with ARM Cortex-A8 processor & 'Verdex Pro' uses Marvell PXA270 with XScale. Temperature sensor (Thermocouples or Thermistor), Humidity sensor & Wind sensor can be connected thro' GPIO/I2C/UART/USB etc to any of the above said mote to run the developed system as it is designed to work as a Weather station detecting all three different parameters mentioned.

4. SOFTWARE IMPLEMENTATION

The Query system software is developed using Java language. A Java Virtual Machine (JVM) can be setup on the gumstix which would run the Java code on the Linux operating system. JamVM(6) is an example of a virtual machine used frequently. It is small in size and complies with Java Virtual Machine(JVM) specification.

The software system is comprised of classes that help the system to be operated in a desired manner. In the developed system, different Sensor's class generates arbitrary values for respected sensor type to represent temperature, humidity & wind speed using a random generator Java component. The Sensor node class collects data from the sensor class and processes the information that is collected. Data is stored at the Sensor node in arrays and same data is replaced with new data every 10 seconds & data's older than 24 hours are over written on FIFO (First in First Out) mode.

The Base station class provides operator three different types of queries to be made on three different sensor nodes & each having three different sensor types. The query on either one of the parameters (temperature/humidity/wind speed) with specification of time range (start time & stop time with respect to present time) for the parameter's average, maximum and minimum can be extracted respectively. When the operator makes a query the base station class enquires the Gateway node class for response to the query. The computed data is returned to Base station and in turn shown to the operator through a graphical user interface (GUI). GUI source code for Client lies in the Client GUI class (node input) & output GUI source in Server class which are constructed to handle all the input and output of the system. The GUI of base station (Input) & Query server (to see Output) is shown below.

Figure 2 GUI of Base Station & Query Server

4.1 DESIGN & WORK FLOW OF THE QUERY

Base station forms the input for the query through GUI provided; the query then from the client is wirelessly forwarded to the Gateway node for further validation of the query format & processing is done to obtain result.

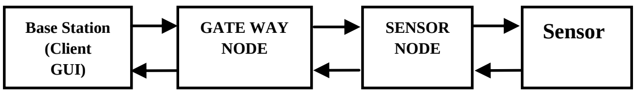

Figure 3 Function analysis of Base station

Based on the above function analysis the structural flow & classes involved for Query to be executed from (Base station point of view) is show below in the next page.

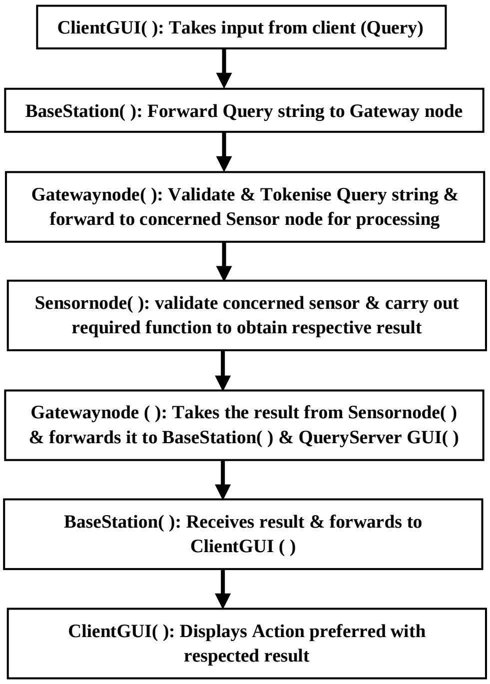

Figure 4 Structural Work Flow of Query

4.2 METHODS & FUNCTIONS OF THE *GatewayNode* CLASS

Gateway Node forms the entry point to the server, where the query through Base station is received through socket connection. Gate way node parses the query & then validates the sensor node specified in it, thus forwarding further query to be processed to the concerned sensor node.

Table 1 Methods & Functions of *GatewayNode* Class

SI No	Method	Function description
1	*ServerSocket ( )*	To listen query from Base station.
2	*ExecuteQuery ( )*	To execute which sensor node to be selected based on query
3	*Sn1 ( )*	To select Sensor Node-1.
4	*Sn2 ( )*	To select Sensor Node-2.
5	*Sn3 ( )*	To select Sensor Node-3.
6	*Validate&TokeniseQueryString ( )*	Forward's to concerned Sensor node for further processing
7	*ServerGui ( )*	To display final queried result on server GUI

4.3 WORK FLOW OF GATEWAYNODE CLASS

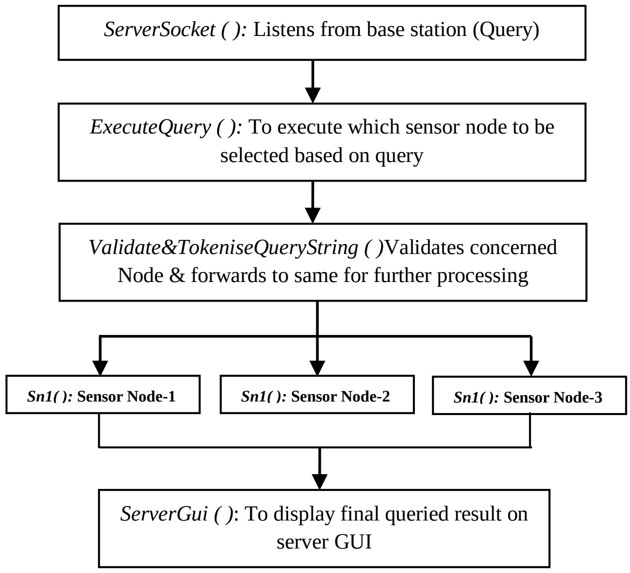

Figure 5 Work Flow of Gateway Node

4.4 METHODS & FUNCTIONS OF THE *SensorNode* CLASS

Methods & functions of SensorNode Class which receives query validate Sensor & processes on query are given below in Table. 2.

Table 2 Methods & Functions of *SensorNode* Class

SI No.	Method	Function description
1	*ActivateSensors ( )*	To activate concerned sensor required to process query.
2	*ReportTemperatureData ( )*	Obtain data from Temperature Sensor.
3	*ReportHumidityData ( )*	Obtain data from Humidity Sensor.
4	*ReportWindData ( )*	Obtain data from Wind Sensor.
5	*ProcessQuery ( )*	Process data in order to achieve query related result
6	*StopSensors ( )*	To Stop sensors that are running

4.5 WORK FLOW OF SENSORNODE CLASS

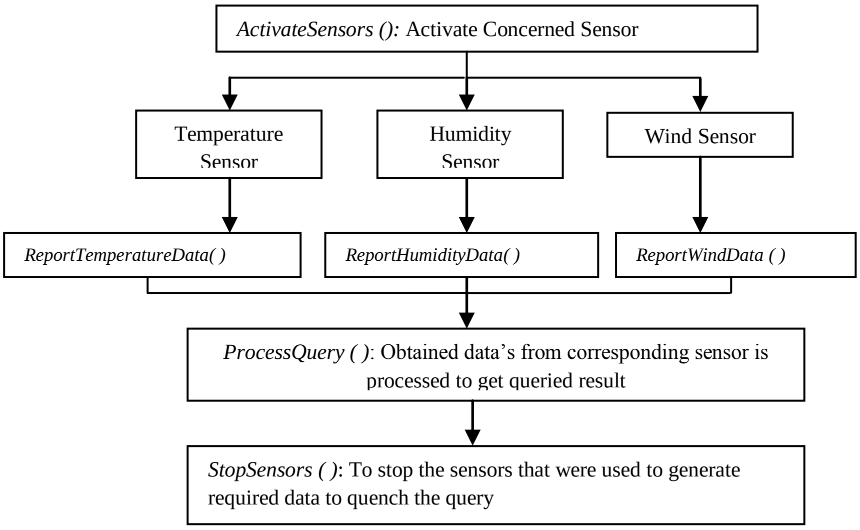

Figure 6 Work Flow of SensorNode Class

5. QUERY FORMAT & EVIDENCE OF SYSTEM PROCESSING

As explained in the previous sections, the proposed system contains three different sensor node's which further is connected to three different types of sensors for sensing three different parameters. To extract a particular data from any individual sensor & processing it calls a need for a specific path for the system to reach for it & this should be specified by the user. A specifically designed Query Format is used in this system in order to instruct the system & perform required processes as per the query. It is recommended for the users to use the suggested format in order to place any query without which this system is non-responsive & may end up with error messages. The query format consists of five tokens separated by comas as shown in figure below.

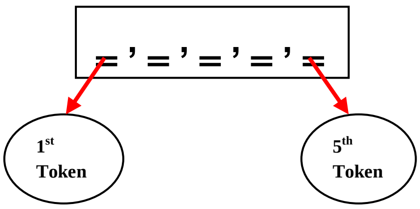

Figure 7 Query Format used

Description of individual tokens & their meaning is given below in Table 3 which is given below.

Table 3 Detailed information of query Format

1st Token	This token specifies the **node number** (ex:- 1 or 2 or 3). As the system has only 3 sensor nodes so only 1 or 2 or 3 can be entered depending on the node required.
2nd Token	This token specifies the **sensor type** in the specified node (ex:- temperature or humidity or wind). For temperature sensor need to enter '**T**'or '**t**' likewise for humidity '**H**' or '**h**' and for wind '**W**' or '**w**'. It is case insensitive.
3rd Token	This token specifies the **start time** in minutes in the past with respect to the current time, i.e. it specifies the starting point of the range of data over which the query should be processed. (E.g. if the user needs to start from 2 Hrs in the past then the value '120' should be entered here.)
4th Token	This token specifies the **time period** in minutes over which the required function will be performed, i.e. the range of data values over which the query should be processed. (E.g. to request a the data over a 60 minute period of time beginning from 2 hours in the past, the third token value should be '120' and this token's value should be 60).
5th Token	This last token specifies the **type of function** to be processed (ex:- '**AVG**' for average, '**MAX**' for maximum and '**MIN**' for minimun). This is case-insensitive. For example if a user wants to know the maximum temperature from node 1 and starting from 2 mins in the past and 1 min after that, then the user should enter in the query in a format like '**1,T,2,1,MAX**'.

A Query with the mentioned format as explained above is shown in the figure 8 with its details explained after the figure 8.

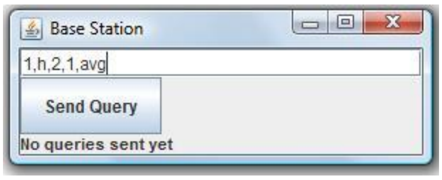

Figure 8 An example of query in previously explained format.

The above figure shows an example know the average humidity of sensor node 1 over a period of 1 minute starting from 2 minutes in the past.

5.1 EVIDENCE OF QUERY BEING PROCESSED TO SENSORNODE

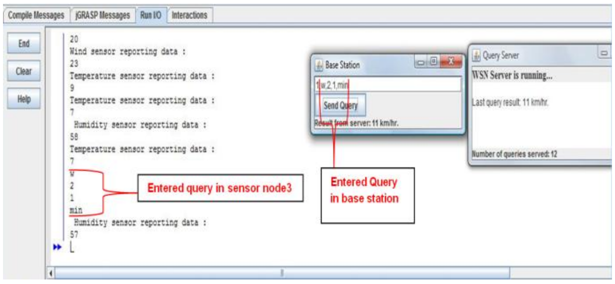

Figure 9 Evidence of query being processed to Sensor node.

The above screen shot provides the evidence for the systems work flow. Once the query is entered & the process starts the Gateway node validates the concerned Sensor node which further activates respective sensor. In the above example the minimum wind speed between time period 2min to 1min with respect to present time is queried from 1st sensor node.

5.2EVIDENCE OF SENSOR REPORTING TO SENSOR NODE IN EVERY 10 SECONDS

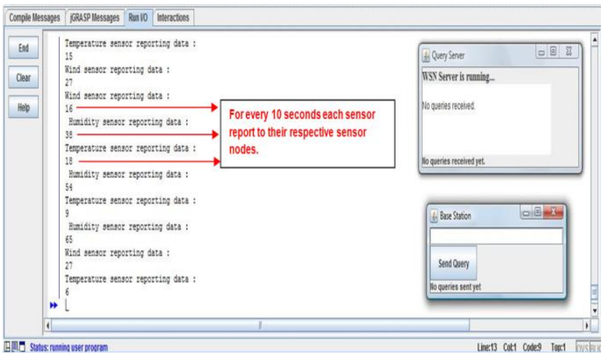

Figure 10 Sensor reporting the data to sensor node every 10 sec.

From Fig 10, it can be seen that every 10 seconds, the sensors Report data to their respective sensor nodes. When query is entered the validation of the sensor node takes place (i.e. sensor node 1 or 2 or 3) in the gateway node and the remaining part of the query is forwarded to specified sensor node.

5.3EVIDENCE OF QUERY PROCESSING

The systems dependability can be judged upon the accuracy with which the system presents the result for the query fetched. So, an example to show the accuracy of the result is shown below which compares the theoretical values with the system presented values taking an example of finding an average temperature value for a time period of 1 minute from a Temperature sensor in Sensor node 1 is considered.

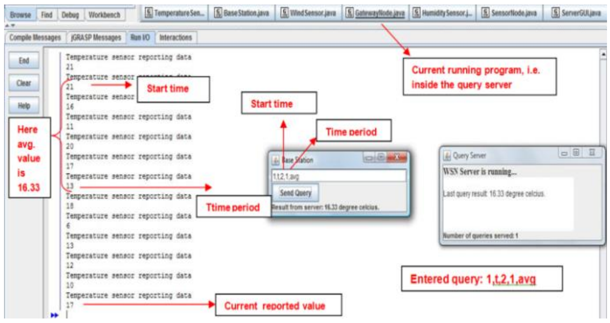

Figure 11 Evidence of Query processing for avg temperature calculation

Theoretical calculations:

Sensor reports value in every 10 seconds
Specified time in minutes, start time = 2min and range in minutes = 1 min.

2min = 120 seconds = 12 values
1min = 60 seconds = 6 values

So according to the query entered, the maximum temperature value can be found by considering first 6 of last 12 values from the current reported value as shown in snapshot.
From snapshot
Current reported value = 17
First 6 of last 12 values = 21, 16, 11, 20, 17, 13.
Average of these values = 21+16+11+20+17+13 = 98,
So, 98/6 = 16.33.

Hence, System presented Values were verified with theoretical values. This shows that this system id dependable.

6. CONCLUSION & FUTURE WORK

This paper presented a software implementation of WSN weather monitoring system successfully. Implementation was done considering some of hardware's available in the market & this system could easily be implemented while using the suggested hardware. The systems versatility was tested considering different aspects which have been noted in previous sections. The performance on the implemented software was evaluated & its real time performance can be only evaluated through hardware implementations. As the first step, the software implementation was considered & successfully completed. The process is on going for its hardware counterpart implementation.

7. REFERENCES

[1] Tinghuai Ma, Hao Cao, Donghai Guan and Sung Young Lee "An Efficient Distributed Weather Data Sharing System Based on Agent" the International Arab J. Of Inform Tech., vol 2, pp. 170-178, March 2012.

[2] Jinbiao Huo, "Research on Design of a Query System of Weather Information in a Mobile Telephone based on MIDP", in Proc. WISA, 2009, pp. 309-311.

[3] David Malan, Thaddeus R.F. Fulford-Jones, Matt Welsh and Steve Moulton "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care " Internet: http://lcawww.epfl.ch/luo/WAMES%202004.htm> , 2004 [Jun. 23,2013]

[4] T. Sun, B. Peng, "Developing an End to End Wireless Application Using MIDP and J2EE," Microcomputer Information, 2006, pp. 159-161.

[5] C. Li, J.R. Luo, H.Z. Wang et al, "Program Composition Method Summary on CLDC/MIDP Client," Computer and Modernization, 2005, pp. 5-10.

[6] Matt Welsh and Geoff Mainland. Programming sensor networks using abstract regions. In the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04), March 2004.

[7] Alec Woo, Terence Tong, and David Culler. "Taming the underlying challenges of reliable multihop routing in sensor networks", In the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), November 2003.

[8] V. Kumar, A. Pentland, "DiaBetNet: Learning and Predicting for Better Health and Compliance", Diabeties Technology Society meeting, Foster City, CA Oct 31-Nov 2

[9] L. Li, I. Harrocks, "A Software Framework For Matchmaking Based on Semantic Web Technology" in Proc. WWW Conference, 2003

[10] V. Agarwal et al., "Synthy. A System for End to End Composition of Web Services" in Journal of Web Semantics, Vol. 3, Issue 4, 2005