



A Heterogeneous Middleware Architecture for Wireless Sensor Network

Bhagwan Sahay Meena
Department of Computer
science
Assam University, Silchar
Assam, India

Neeraj Jain
Department of Computer
Science and Engineering
Scope College of Engineering,
Bhopal, India

Chhaya Singh
Department of Information
Technology
Scope College of Engineering,
Bhopal, India

ABSTRACT

Wireless Sensor Network (WSN) consists of tiny devices, called sensors [1] are used to observe physical environmental conditions, such as temperature, light, pressure, sound, speed and so on. Wireless Sensor Network plays a significant role in many applications like monitoring of Environment and Habitat Conditions, Medical applications, Military Surveillance, Traffic Control, Civilian applications, etc. The number of information sources has rapidly increased the need for intelligent mediation tools to be implemented between the users and resources [20]. The main problem of information source integration in their heterogeneity at the level of their format. It is not possible to change these existing sources in order to make them homogeneous (for example XML or relational). To build architecture on the top of these sources can solve the problem and allows querying an information system which is centralized and homogeneous. In this Paper we designed architecture for heterogeneous wireless sensor network, which accepts multi queries in XML format, known as XQuery. The user uses XQuery instead of SQL. This paper proposes architecture to remove heterogeneity in term of database from wireless sensor network.

Keywords

Wireless sensor network, sensor mote, XQuery, SQL.

1. INTRODUCTION

WSN is a group of several wireless sensor nodes that connected to the base station node by wireless medium. These tiny sensor nodes measure environmental and physical properties like temperature, humidity, pressure [1, 2] and organize themselves to deliver this sense data to a common data collector base station over the wireless medium. Sensor nodes sense data by using the sensing unit and sensed data processing of analog to digital unit. This unit digitizes the analog data into the digital and after that the digitized data send to the microcontroller unit. Now this unit sends the data to a base station for further processing. A sensor node is a low cost small hardware device suited for large scale deployment and autonomous operations. The cost, physical size, operational requirements, deployment regions and wireless communication medium restrict the processor capacity, memory size, energy, operating platform and reliability of the sensor nodes and the sensor network.

The sensor nodes in the network can be considered as a network database [2]. The database of the sensor network distributed over the sensor nodes, this data retrieve by user with the help of queries. The location of data sensed by sensor nodes should be significance. If the location of the event

occurrence observed by a sensor node is not known, the sense data about the event does not be meaningful. Moreover, the data reported by sensor nodes are largely raw. Thus, there is a need for a query system [10, 24] and distributed query processing [24, 26] that requires sophisticated processing of data generation and storage of distributed data. A WSN is limited in power, communication, storage etc. Hence, other ways have to be devised for querying, data gathering and other unique requirements. Due to the limitation of sensor node size, The TinyOS [17] is used for operating the nodes. This operating system basically used for only sensor motes. As we know the sensor nodes have limited energy resources. Each node plays an important role in the communication for data transmitting as well as data receiving. So, the batch Query system is not energy efficient. So the traditional sensor network degrades the energy lifetime of the sensor nodes.

2. LITERATURE REVIEW

In the wireless sensor network multiple queries are resource-constrained. It is critical to execute them efficiently because WSN has limited resources. A middleware [8, 9, 10, 11] is used as an interface between high-level abstraction and system-level programming concept in WSN. It can be used in such applications to hide the heterogeneity and distributed nature of the networks. The middleware [17, 18] is software that can be used to fill the gap between low level constructs and the user applications. A crucial middleware used in query processing in WSN is TinyDB [7]. TinyDB provides a single unified declarative query (such as SQL) interface for the ease of communication between user and sensor networks. A user can fire Tiny-SQL queries (similar to SQL queries) to the base station. These queries are distributed to all the relevant nodes in the WSN from the base station that gather the result and send the response back to the user. Due to limited resources, Tiny-SQL does not support all types of SQL queries. The traditional TinySQL has several limitations. It does not give optimized result in multiple query scenarios. TinyDB uses a distributed network query processing architecture. The components of the TinyDB are built on top of TinyOS [26]. The user query should pass through these entire components [3].



Table 1. Compression of Middleware architecture

Design Principles	Sina	Tiny Lime	Mires	Milan	Mate
Energy Efficiency & Resource Mgmt	Y	Y	Y	Y	Y
Data-Centricity	Y	Y	Y	Y	N
QoS Support	N	N	N	Y	N
Application Knowledge	N	N	N	Y	N
In-Network Processing	Y	N	Y	N	N
Scalability	Y	N	Y	Y	N
Adaptability	N	N	N	Y	N
Security	N	N	N	N	Y
Dynamic Net Topology	Y	Y	N	N	Y
Configurability & Maintainability	N	N	N	N	Y
Heterogeneity	N	Y	Y	Y	Y
Real World Integration	Y	Y	N	Y	Y

3. PROPOSED ARCHITECTURE

The traditional query processing in WSN uses SQL query interface. The user sends Tiny-SQL queries to the server. Due to limitation in SQL, the query processing system has three main limitations: i) SQL works well with the relational model and hence a SQL query returns only tabular data. The structure of a sensor network is hierarchical and need a query language that can traverse in a tree manner; ii) It is dependent on the underlying components; and iii) this system does not give an optimized performance for multiple queries. The drawbacks present in the system motivated for alternative ways, namely 'XQuery' for execution of the query which is coming continuously and also for the multiple query request with optimization. The advantage of XQuery over SQL is the first step towards the development of XQuery based query engine for query processing in sensor network. The limitation listed above is as addressed by XQuery are as follows.

- XQuery is very flexible in nature. It can query both tabular and hierarchical data; therefore, it is suitable for the sensor network.
- Extensibility – XQuery supports user defined tags depending on the application.
- XQuery is independent of the underlying software.

3.1 XQuery and SQL based Heterogeneous Architecture

This proposed architecture giving a query interface to the user to interact with WSN without worrying about query format. This proposed architecture depends on execution of multiple queries [12] in embedded network. In this proposed architecture user sends queries to the server in XML format,

this type of query known as XQuery [24, 25]. The user also gets the response in XML format from the sensor network. As we know in the traditional network the clients sends the query in the SQL format and also get the response in SQL.

This architecture based on XML query interface instead of SQL inbuilt interface to interact with sensor network by using the Java GUI interface. The proposed architecture is subdivided into different module given below.

3.2 Working of Modules

The proposed engine is divided into three modules as client side, base station and wireless sensor network. The functionalities of each module are as follows:

3.2.1 Client or user side

This module is the client side module that contains all clients. According to the above architecture there are N numbers of clients that send the query in XML format and also get the response in XML format. Client sends Query, which goes to the server (base station) and the server sends it to sensor network for retrieving data. The server receives the data from sensor networks and sends response to client in XML format. At the client side, the Query always is XQuery that send to server and also receive the XResponse from the server.

To solve the heterogeneity problem in term of database from sensor nodes, we have defined some input parameters (node id) for the user. By using these input parameters the client sends XQuery to the server. The client does not have any idea about that what type of data exist on the sensor nodes, whether which node having SQL data or XML data. The client just selects the input parameter and sends the XQuery to the server.

3.2.2 Base Station

This module acts as an intermediate between the client module and wireless sensor network module. The base station module configured on the high processing machine, which play the role of server for this architecture and performs some main function in the whole scenario. It performs following tasks in two cases.

Case 1: The input parameters taken by the client in the XQuery examine at the server side. If the client wants some information from the sensor network and information is in relational database then the following step takes place at the server side for parsing XML to SQL and also vice versa.

It receives a client request in XQuery form.

- Convert the XQuery request into the corresponding SQL Query.
- The converted SOL query sent to the root of WSN.
- It receives the response from WSN.
- Received SQL response converts into the XResponse.
- XResponse send to the corresponding client.

According to our given architecture we divided the base station module into four sub modules for convince, each module describe below:

3.2.2.1 XQuery Listener

The function of this module is to listen XQuery from the clients that are connected to base station.

3.2.2.2 XQuery Parser

The function of this module is that it performs parsing of the Query received from XQuery Listener. The XQuery converts to corresponding the SQL Query that send by the XQuery



Listener and after the parsing, it sends to the wireless sensor network for retrieving the information.

3.2.2.3 SQL parser

The main function of this module is to perform parsing of SQL data. It retrieves the information from the sensor network in SQL form and parses this information from SQL to XML format. After parsing, the XResponse send to the client. The parsing takes place at the cygwin thread class.

Case 2: If the client wants to retrieve information from the sensor notes that are in the XML format then the client sends XQuery directly to sensor networks root node by XQuery Listener.

3.2.2.4 XQuery Response

The information retrieved from sensor notes in XML format received at the root node. The root node will send this information to the XQuery Response and this XResponse will send to the client .The data available on the sensor mote in XML format, so there is no need of query parsing.

3.2.3 Wireless Sensor Networks

The whole WSN is organized as a hierarchical network with multiple layers and each layer is having one or more parent sensor notes. One mote elected as root node responsible for taking the SQL query as input and passing the same to each child sensor notes as shown in figure 1. There are two group

of sensor nodes, one contains relational database (in above circle) and another contains XML database (in below circle). Each sensor mote creates a data packet containing the node ID and sensed data. The node ID of root node is '0' and other nodes have node IDs 1, 2, 3 ...n etc.

4. EXPERIMENTAL SETUP

In this proposed architecture we used MICA2 (MPR400) motes as sensor nodes. First of all, we assigned ID to each node by using the programming board MIB510CA. The ID of the root node should be zero (0) and id of other motes are 1, 2, 3---n.

4.1 Process of Assigning ID to Motes

We had taken one sensor programming motherboard MIB510CA and MICA2 motes and one serial to serial RS-232 cable and one 5 to 10 volt DC supply adaptor. One end of serial 9 bit connector connects to base station PC and the other end to the programming board. The DC supply is also connected to the programming board. Now we have opened the cygwin window [26] and type:

```
write -> make mica2.0, /dev/ttyS0 {it is for root node}
```

Here make: for compiling the code.

mica2.0: is platform and assign ID. / It may be telosb.

/dev/ttyS0: it is the port number (COM1).

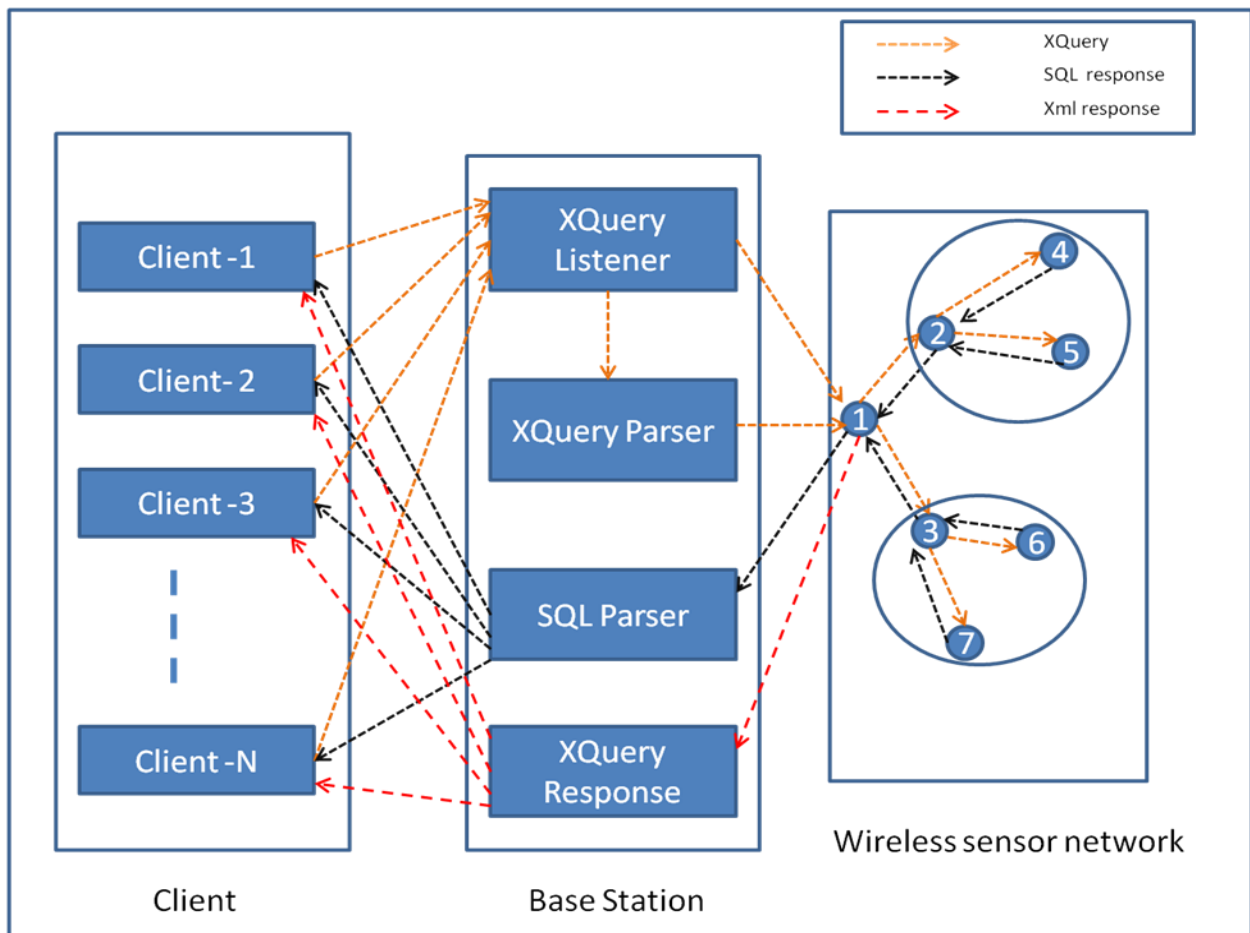


Fig 1: The Proposed Architecture



After assigning an ID to the root node, the mica2 mote removed from the programming board and other motes are connected one by one. We have written the install.1, install.2, install.3 up to install.n for other motes respectively. After the whole process, each mote has an ID and then deploys these nodes in sensing field. Now we again connect to the root motes to the programming board and install another application.

4.2 Client Side Implementation

In this phase, we describe the client and server communication process. We have designed two classes on client side, first is client class and the second is client thread for multiple clients. First we start client side GUI that will connects to the server using IP address and port number. In this implementation server has started on port (9876) and send startup message to client side. Now client sends request on the given port number and IP address and it will also set timer [time out (ms)] for a request. If connection established successfully then server send an ACK to the client .Now the client sends an XQuery in XML format and wait for server response.

4.3 Server Side Implementation

On the server side (base station) implementation we designed six classes. In this implementation multithreading concept is used for handling the client XQuery. When the server listens to the client Query it spawn a thread automatically and information will process.

Now client Query comes to server side and server send received Query to root node according to selected input parameters. Server retrieves the result form sensor network and sends back this result to the client. In the figure 1 we can see the data retrieved from relational database and converted to XML response. Finally the server sends this data to client side in XML response as shown in figure 2.

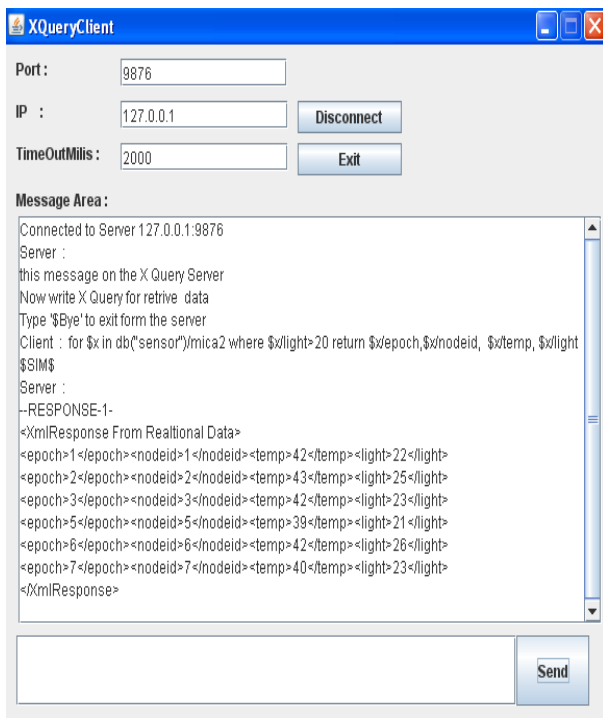


Fig 2: Data received at client side

4.4 The Process of Proposed Work

The process of proposed work subdivided into three stages.

4.4.1 Stage 1

In the first stage, we have included client and server communication via socket programming as well as multithread programming for client server communication. For this we used java (jdk 1.6version) language and net beans (6.0 version) tool that provides GUI interface. By using java we established multithread communication between the client and server.

4.4.2 Stage 2

In the second stage we convert the XQuery to the SQL query by parsing at the server side, if the input parameter belongs to the relational data otherwise the XQuery directly send to the root mote .All these function are performed by XQuery Parser. The XQuery Listener Listen the client request and parsing takes place at the XQuery parser. It simply converts the XQuery syntax to the SQL syntax. The parser sends SQL query to the root node of sensor network and root node retrieve the relational data from database.

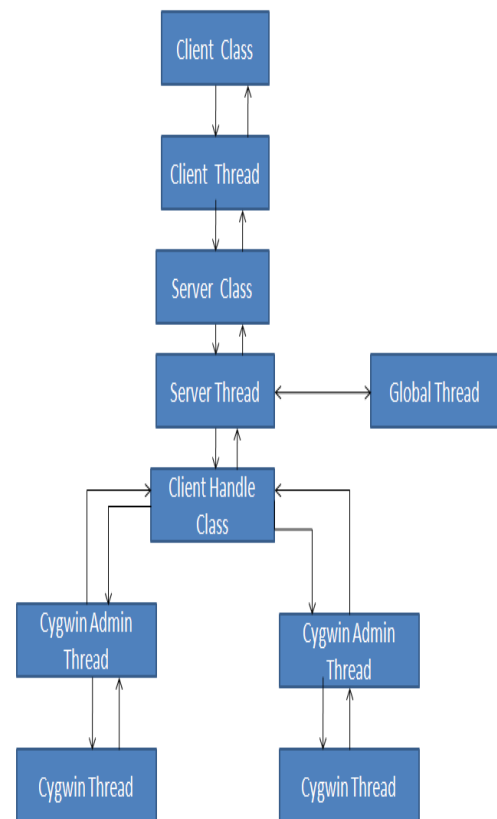


Fig 3: Flow chart of proposed work

4.4.3 Stage 3

In the last stage the base station and sensor network communication takes place. Now, we have two types of nodes as shown in figure 1, one group of nodes having relational type of data and second group of nodes having XML type of data. The parsed SQL Query fire on those motes that are



having relational data. The root node retrieves data which is relational type. Now root sends data to the server. Once again the SQL data parse into the XML and forward to the corresponding client. For the second types of data which is in XML format the motes. XQuery received at root mote, send this XQuery to all motes that having XML data. For retrieving data from these motes we can directly send XQuery to the motes and retrieve data at the root node. Here we stored data in XML format on these motes. The root node sends this data to XQuery Response which sends this data to the corresponding client.

5. ANALYSIS OF PERFORMANCE

We proposed this architecture to remove the heterogeneity of database from sensor motes and increases the lifetime of the sensor motes. As we know in the TinyDB, the user can send only a single query to TinyDB at a time in SQL format and wait for response. After getting the response of previous query user can send the next query. Using proposed architecture a user can send multi queries in XML format and also receive response in XML. So users can send XQuery instead of SQL query in the sensor network and also use the new feature of XML. The SQL query did not give the optimized result in sensor network but the XML query gives the optimized result in the sensor network.

The table 2, 3 and 4 shows the data retrieved from mica 2 motes for different-different parameters like as temperature, light and on the basis of them the data will be collected in tabular.

Table 2. Data retrieved from Mica2 mote-1

Epoch	light	temp
1	22	42
2	23	43
3	26	42
4	24	40
5	25	38
6	28	41
7	26	40

Table 3. Data Retrieved from Mica-2 Mote-2

Epoch	light	Temp
1	25	43
2	21	39
3	27	37
4	25	36
5	27	38
6	29	35
7	30	37

Table 4. Data retrieved from Mica2 Mote- 3

Epoch	temp	light
1	40	23
2	38	25
3	41	25
4	39	26
5	38	29
6	37	27
7	36	28

The data received from mica-2 motes are different when user fires the XQuery on different-different types of database such as relational (mote 2) or XML (mote 3). The above table is designed after retrieving the data from motes. The data received from motes are aggregated at the user side. So according to the ID of each mote the data are collected.

The proposed architecture provides a new GUI environment for the user instead of TinyDB GUI and also provides the optimized result.

6. CONCLUSION

In this paper we proposed architecture for removing the heterogeneity from the sensor network in term of database and also prolong the lifetime of the sensor nodes. It is also supports multi-query environment instead of a single query. The SQL query cannot optimize output and also not gives feasible results. In this paper we used an XQuery processing technique instead of the SQL query on the client side. In this architecture the client sends XQuery in XML and also receives the response in XML. The paper removes heterogeneity problem from databases. If the data is in relational form then parsing takes place and if the data in XML form then the XQuery directly fire on the sensor nodes and get the response in XML. This paper provides XQuery concept, reason being in the tree base network. The main advantage of this paper is that: first it removes heterogeneity and second it increases the lifetime of the sensor nodes.

7. FUTURE WORK

This paper can be extended by implementing metadata management by using XML data. By using the XML data, the node takes less power in query processing. The central XML data management increases the reliability of data on mote.

8. REFERENCES

- [1] Heidemann J., Fabio Silva F., Intanagonwiwat C., Govindan R., Estrin D. and Ganesan D., "Building Efficient Wireless Sensor Networks with Low-Level Naming", In Proceedings of 18th ACM Symposium on Operating Systems Principles, 2001.
- [2] Govindan R., Hellerstein M., Hong W., Madden S., Franklin M. and Shenker S., "The sensor network as a database", Technical Report-No. 02-771, 2002.
- [3] Akyildiz I.F., Su W., Sankarasubramaniam Y., and Cayirci E., "A survey on sensor networks", IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, 2002



- [4] Ramesh Govindan, Joseph M. Hellerstein, Wei Hong, Samuel Madden, Michael Franklin, Scott Shenker, “The Sensor Network as a Database”, No. 02-771, September 2002. [Technical Report]
- [5] Yong Yao Johannes Gehrke, “Cougar Approach to In-Network Query Processing in Sensor Networks” Department of Computer Science Cornell University.2002
- [6] David Gay, Philip Levis, Robert M. Welsh, Eric Brewer, David Culler “The nesC Language: A Holistic Approach to Networked Embedded Systems”, University of California, 2003
- [7] Madden S., Hellerstein J. and Hong W., “TinyDB: In-network query processing in TinyOS”, Version 0.4, 2003.
- [8] Yong Yao, and Johannes Gehrke, “Query Processing for Sensor Networks”, Department of Computer Science, Cornell University, In CIDR, 2003.
- [9] Sam Madden, Joe Hellerstein, and Wei Hong, “TinyDB: In-Network Query Processing in TinyOS”, {madden,jmh}@cs.berkeley.edu, whong@intel-research.net, Version 0.4, September, 2003
- [10] Yao Y. and Gehrke J., “Query processing in sensor networks”, In Proceedings of International Conference on Innovative Data Systems Research, 2003.
- [11] Ville St. L. and Dickman P., “ Garnet: A Middleware architecture for distributing data streams originating in wireless sensor networks”, In Proceedings of 23rd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 235-241, 2003.
- [12] Johannes Gehrke Cornell, “Query Processing in Sensor Networks”, University Samuel Madden MIT, 2004
- [13] Heinzelman B.W., Murphy A., Carvalho H., and Perillo M., “Middleware to support sensor network applications”, IEEE Network, vol. 18, no. 1, pp. 6-14, 2004.
- [14] Woo, A., Madden, S., & Govindan, R. “Networking support for query processing in sensor networks”. Communications of the ACM, 47(6), 47-52, 2004
- [15] Souto E., Guimaraes G., Vasconcelos G., Vieira M., Rosa N. and Ferraz C., “A message-oriented middleware for sensor networks”, In Proceedings of 2nd ACM International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), pp. 127-134, 2004.
- [16] Fok C., Roman G. and Lu C., “Mobile agent middleware for sensor networks: An application case study”, In Proceedings of 4th International Conference on Information Processing in Sensor Networks, pp. 382-387, 2005.
- [17] Hadim S. and Mohamed N., “Middleware challenges and approaches for wireless sensor networks”, IEEE Distributed Systems Online, vol. 7, no. 3, pp. 1-23, 2006.
- [18] Hadim S. and Mohamed N., “Middleware for wireless sensor networks: A survey”, In Proceedings of 1st IEEE International Conference on Communication System Software and Middleware (Comsware), 2006.
- [19] Hadim, S., & Mohamed, N., “Middleware: Middleware challenges and approaches for wireless sensor networks”. *Distributed Systems Online, IEEE*, 7(3), 1-1, 2006
- [20] Ezziyani, Mostafa, Mustapha Bennouna, and Loubna Cherrat. "An advanced XML mediator for heterogeneous information systems based on application domain specification." *International journal of computer science and applications* 3, no. 2 (2006).
- [21] Xiang, S., Lim, H. B., Tan, K. L., & Zhou, Y., “Two-tier multiple query optimization for sensor networks”. In *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on* (pp. 39-39). IEEE.
- [22] Yoon S. H. and Shahabi C., “Distributed Spatial Skyline Query Processing in Wireless Sensor Networks” In IPSN, 2009
- [23] Krishnakumar T., “Integrated Routing and Query Processing in Wireless Sensor Networks.” *International Journal of Applied engg. Research*, Vol. 1, No. 1, 2010.
- [24] Da Silva R.I., Del Duca Almeida, V., Poersch A.M., Nogueira J.M.S., “Spatial query processing in wireless sensor network for disaster management” *Wireless Days (WD)*, 2nd IFIP, pp. 1-5, 2010.
- [25] Kumar, Manish, Monalisa Panigrahi, and Shekhar Verma. "XQuery based Query Processing Architecture in Wireless Sensor Networks." *International Journal of Computer Applications* 43, no. 23 (2012): 5-10.
- [26] www.csl.stanford.edu/~pal/pubs/tinyos-programming