TikiriDB: Shared Wireless Sensor Network Database for Multi-User Data Access

Nanayanajith M. Laxaman, M. D. J. S. Goonathillake, Kasun De Zoysa.

University of Colombo School of Computing

No. 35, Reid Avenue, Colombo 7, Sri Lanka

{nml,jsg,kasun}.ucsc.cmb.ac.lk

ABSTRACT

With the rapid development of embedded systems, the use of wireless sensor networks (WSNs) has been increased among many private and government organizations. Other than them there are individuals and communities such as biologists, ecologists, tourists, climatologists, who interest in sensory data. However, all interested parties may not be able to deploy such networks on their own. Because, deployment of a sensor network is expensive, there may be government regulation restrictions on deploying sensor networks, there can be accessibility issues for certain sites. A feasible approach is to share a single WSN by several communities to achieve their individual goals. Then, these networks can be deployed for the benefit of organizations and can also be open to the non technical. Hence, there are issues on giving access to all communities to a shared WSN. In this paper we propose a comprehensive solution called TikiriDB to overcome these issues.

1.0 INTRODUCTION

The research in WSNs and successful solutions provided through it to industry has made it popular around the world. WSN have initiated the digitization of sensing phenomenon in nature, industry, human-body and many other natural environments and artifacts. Currently, most countries use WSNs to facilitate and deploy automation solutions for areas such as agriculture, biology, public and health services.

Currently WSNs can be deployed to sense many factors such as temperature, humidity, gases, location, proximity, sound, video, light, moisture. In our research we are interested in shared WSN. A few examples for such WSN applications are: Medical WSNs allows doctors to remotely monitor patients' conditions in real-time to provide a quality health service [1]. A WSN which facilitating the decision making process of the traffic police officers to solve traffic congestion. All such WSNs are deployed by one authority and should be accessed by many users and such WSNs are considered as shared WSNs, whereas if we consider the medical sensor network, many doctors will access the same sensor network deployed to measure patients' conditions.

Even though multiple user access to the WSN beneficial to the individual users, there are issues to be solve before opening the WSN for users. First, the data requests should be easily customized or altered according to the user requirements. The language used to request data should be easily understandable to both users in WSN discipline and to user in other disciplines. Second, it should support multiple query points, multiple users with concurrent access, and pacify simultaneous data requests even from different geographical locations. Third, there must be a reliable access control mechanism implemented to control accessibility to the data and resources of the sensor nodes. Finally, since, WSN deployments generally use dried batteries as the power source it must be power efficient for the long run.

Therefore, we propose TikiriDB to solve above mentioned issues. It is a shared WSN Database system which can be manipulated easily through simple queries. It is a variation of standard Structured Querying Language (SQL) which adopted and optimized to the sensor network environment [2]. The query processor and the routing protocol of TikiriDB are designed to facilitate acquiring data by many users simultaneously from any point of the WSN. Furthermore, a distributed access control mechanism using public-key infrastructure was incorporated with TikiriDB. Hence, the proposed solution can cover most of the problems arose in a shared WSN environment.

The semantics of the queries used in TikiriDB are similar to the TinyDB's[2]. A simple example of a query can be given as follows.

```
SELECT temp, humid FROM sensors
SAMPLE PERIOD 2 FOR 10
```

This query returns node id, humidity level, and temperature level in every 2 seconds intervals for duration of 10 seconds from all the available sensors nodes in the sensor network. The result of this query has been illustrated in Figure 4. The resulting table is dynamically expanding according to the time and results will be appended to the end of the table as they are arrived to the user. That is, the result set is automatically sorted by time. To prevent multiple query execution for similar sensor data acquisition, a multi-query optimization was proposed adjacent to the query processor.

The routing protocol of the system is specifically designed to meet the requirements of TikiriDB. The routing protocol is based on multi-casting. When considering sensor networks, even if multi-casting is not widely investigated area, it can be used to save energy by minimizing the network usage by reaching multiple destinations without having to broadcast messages [3]. The TikiriDB routing protocol enables access to sensor network from any point of WSN without any performance reduction. Users only need to have an authorized base-station with them. Then it is possible to connect with the network through any node, and collect sensor data from any remote location where he/she can access the WSN. The system is designed in such a way that the user not necessarily have to be in the same location to get the results of his/her query i.e. injecting query from one point to WSN and receiving results from another point from WSN. We refer this as base-station mobilization.

To identify different authority levels and control query execution by users, query processor incorporates an access control module. This module can be enabled or disabled according to the requirements. The access control module will verify whether the user possess sufficient authority before executing a query. In addition to that, the integrity of data and encrypted communication mechanisms has been considered to increase the confidentiality.

The remainder of this paper is organized as follows. Section 2 brief our approach and in section 2.1 gives the overview of what a shared WSN is. Section 2.2 gives the design of our query processor. Section 2.3 reviews routing protocol and section 2.4 illustrates the access control module. Finally, section 7 concludes our discussion.

2.0 OUR APPROACH

TikiriDB is a complex project which is composed of several components. Mainly a query processor, routing module and an access control module. Each component is expected to be covered with separated concept papers in near future. In this section, we discuss the total architecture and conceptual design which combines these components to form TikiriDB. The implementation of the TikiriDB was done on top of Contiki platform [4].

2.1 Shared WSN

We identified, there are three main elements in the proposed communication architecture; WSN, base-stations, and the users. WSN can be composed of geographically dispersed non-isolated sensor nodes. One or more base-stations can facilitate one or more users to communicate with the WSN in static or Ad-Hoc mode. There are two categories of users in the system, owners who deploy and maintain the sensor network and the users who request for sensor data. Four possible scenarios of these three elements can be found and the interpolation between our shared sensor network design and these scenarios can be detailed as follows.

When there is only one user in WSN, then the scenario can be illustrated in Fig 1. In Fig1.b multiple users connect to the WSN through one base-station. This type of scenario occurs when the single base-station is connected to a pool of user devices such as PCs. Then the queries are routed through the common base-station which acts as the gateway to the WSN. Fig 1.c and Fig 1.d illustrates a scenario where multiple users connect to WSN through multiple base-stations. Since, there are multiple users in scenarios illustrated in Fig. 1. [b, c, d], they can be considered as shared WSNs.

TikiriDB is designed to supports the shared sensor network scenario illustrated in Fig 1.d. Thereby; it will also work for all the other shared sensor network scenarios illustrated in Fig 1.b through Fig 1.c. as well as the unshared scenario illustrated in Fig 1.a.

2.2 Query Routing

Routing is a core component in distributed WSN systems. Even in TikiriDB, it is important and critical to reliably route user injected query to specified destinations. Afterward, the system must route back corresponding results to requested users. Therefore, in a shared WSN, multiple user communication must be precise and accurate. Multi user communication becomes very difficult in situations where users are moving while reading and making requests to WSN. Then intermittent connectivity between nodes and base-stations must also be handled. However, network transmissions are the most power intensive functionality in sensor networks [2]. Further, changing batteries of nodes in a WSN is very difficult and expensive. Therefore, it is important to keep network transmission to a minimum level while achieving required functionality of the system.

According to the query, it can be destined to rout either to a single node, to a selection of nodes or to all the nodes in WSN. Uni-casting and broadcasting protocols are available and established for sensor networking environments which facilitate the queries destined to single node or to all the nodes [4]. When it required destining the query to a selected set of nodes it is possible to use uni-casting or broadcasting protocols. However, multi-casting is more efficient in the perspective of our system. Because, multi-casting is all about delivering a message to intended group of nodes. Hence, network traffic can be minimized [3].

In addition to that, WSN may compose of nodes heterogeneous functionality and resource possessions. For example, MICAz motes have 8MHz processor while the processor speed of Imote2 can have up to 416 MHz. Therefore, when the routing protocol builds up the infrastructure, different sensor nodes may eligible to become a root, to become a node or to become both. A tree is composed of many sensor nodes and a root node. A root node for each tree is elected from the nodes and base-stations by considering available resources such as memory, processor and battery power and considering the direct connectivity to the back bone. A root node is the richest resource holder in a particular tree of nodes. Therefore, most of the time, a base-station becomes a root node for a tree. The back bone of WSN network infrastructure is formed by interconnecting roots of individual trees using graph or tree network topology. When a query is injected to the network through a base-station, it propagates through the back bone and finds the root of the tree which holds destination node.

The root election process and networking infrastructure building process must be initialized by the authorized owner of WSN. The owner can turn the WSN in to sleeping mode or functioning mode with sending simple control queries. The articulation of new nodes is handled by the regional trees at the instant it comes to the vicinity. Figure 3 illustrates proposed routing tree topology.

2.3 Distributed Access Control

Access controlling is important to function a multi-user shared WSN due to the existence of data and resource with different values, and users with different disciplines. Mainly, there are three possible implementations of access control. First, the access control can be implemented in querying base-station itself. Since, user keeps it with him/her and has physical access to the particular base-station, after installation of access control layer in to it the user may attempt to alter application to access unauthorized data. Therefore, implementing access control within querying base-station itself is problematic.

Secondly, access controlling can be implemented in a centralized architecture where a special base-station belongs to WSN is used to provide access controlling service [5]. An injected query is routed to this special base-station and then the base-station checks the authorization of the query. If authorized, the query will be processed within WSN to provide results. However, this architecture is prone to single point of failure [5].

The total functionality of WSN is dependent on the access control base station. In addition to that, the nodes neighboring access control base station are prone to saturate by network traffic or exhaust.

Finally, access controlling can be implemented by using a distributed architecture where access control functionality is implemented inside sensor nodes. Most of the issues identified in above two architectures can be avoided using distributed access control architecture.

Access controlling involves several security concepts. Authenticity, confidentiality, authority and integrity can be considered as the core concepts on controlling access. Hence Public-key cryptography can be used to enable authentication, confidentiality and integrity. Public key cryptography eliminates the complicated key management and pre-distribution required by symmetric key schemes, and provides a very clean interface between the user and sensors [Wang06distributeduser].

To enable authorization an attribute Certificate (AC) is used. An Access Control List (ACL) is included in AC to control the accessibility by resource and data in sensor nodes, such as sensor type, acquiring time, acquiring frequency, nodes' storage limit, nodes' memory limit. The destination node checks ACL before executing any query. If the queries comply with the access granting policy defined in ACL, the query will be executed. Otherwise, it will be discarded and a warning message will be sent back to the user.

Figure 3 illustrates the inside of TikiriDB message. Initially, the user must have obtained a certificate from the Access Control Center (ACC) which is located at the owner of WSN. ACC is the central authority which handles all the PKC and AC related functions except certificate revocation i.e. it work as Certification Authority (CA), Registration Authority (RA) and Attribute Authority (AA). User must possess both PKC and AC from the ACC before injecting any query. All sensor nodes will have the root certificate of ACC at the time of deployment. Sensor nodes will use root certificate of ACC to verify PKC and AC of the user.

2.4 Concurrent Query Processing

Users with their user applications can insert queries into WSN, receive results and handle security activities through their base-station. The queries sent by users will be parsed, checked for errors, preprocessed, and converted into a sensor node readable format at base-stations and sent out to the sensor network. In TikiriDB shared environment, there can be many data request queries destine towards the same sensor node. So the query processor must support concurrency.

To achieve the concurrency in an efficient manner, queries are optimized by creating a global query using individual queries sent by all users during a defined time period. Then the global query is executed. All new queries received at the time of execution of individual query are incorporated with the running global query instantly. The result of the global query is temporarily stored in a buffer to be used by individual queries. Original queries can then fulfill their needs from the buffer and send the data back to the user. For example, if one user requests temperature with a sample period of 10 seconds and another user requests temperature with a sample period of 20 seconds, then the global query will acquire temperature in a sample period of 10 seconds and store them in a local buffer. However, there are some advanced queries such as event queries, network aggregation queries, etc which are optimized in dedicated ways.

The maximum number of concurrent queries can be configurable by the owner of WSN before the deployment.

3.0 RELATED WORK & DISCUSSION

Designing database abstractions for sensor networks has been explored for several years [2][6][7]. TinyDB is the most popular and widely used database abstraction for sensor networks among several other database abstractions [2]. It has introduced and defined a solid SQL variation to be used with WSN. Similar semantics used in SQL variation introduced by TinyDB is used in TikiriDB.

Most of the research related to shared sensor networks, consider sharing resources among internal units and it is a different scenario from

TikiriDB [8][9]. In TikiriDB, sensor network as a whole is shared among multiple users simultaneously. It should also be mentioned that we found several other research very interesting and incorporated promising aspects accordingly to improve multi user capabilities, TikiriDB query processing and query optimization[10][11][12].

Furthermore, we investigated multi-cast routing systems as to elaborate our routing protocol [13][14][3]. Huang, Q et al. focus on a multicast protocol for just-in-time delivery of data [13]. Hsing Feng and Heinzelman and Su et al. focus on energy efficiency of the multicasting protocol [3][14]. These multi-casting protocols are not designed to ease the customization. Therefore, we designed TikiriDB routing protocol to handle mobility and ad-hoc nature of the sensor nodes while making it highly configurable by users to facilitate user decided functionality to preserve energy of WSN.

When considering the security aspect of WSN, there have been a considerable amount of research in the area of securing sensor networks on various aspects [15][16][17][18][5][19]. However, no compatible solution is given to use PKC and AC as hybrid security architecture to control access of the users. Wang, H. & Li, Q. has done a study and has come up with an interesting solution to solve access control problem related to WSN [5]. However, there are significant differences between their research and our research work. First of all, it was not specifically designed for a database system. Secondly, the illustrated protocol in their paper is different from ours. According to the protocol they proposed, we found that it uses more network transactions. Third, they have used two different mechanisms for access controlling local and remote nodes. The remote authentication requires considerably many network transactions. However, network communication is one of the most power consuming actions in WSN. Therefore, we implement our access control module by keeping network transactions to a minimum. Elliptic Curve Cryptography (ECC) is considered to be an effective way of achieving pretty good security due to its fast computation, small key size, and compact signature [16]. For example, to provide equivalent security to 1024-bit RSA, an ECC scheme only needs 160 bits [16]. We opted to use ECC based functionalities adopted from TinyECC to implement public-key based cryptographic functions. We made a good contribution to the sensor network community by porting it from TinyOS to Contiki.

4.0 CONCLUSIONS

Since, the emergence of Contiki, we were trying to find a solution like TinyDB(which is built for TinyOS) for Contiki Operating System. However, the existing research revealed consequences of using certain architectures and technologies. Therefore, we reviewed many ways in which to come up with a sustainable solution for the issue. As a result of that, we came up with a solution which uses shared WSN concept with distributed access control. Initially, we developed TikiriDB for Cooja, the simulator of Contiki OS. However, when trying to port TikiriDB to actual sensor nodes, we came across many issues. As a result of finding solutions for those issues, we contributed to the community with a serial forwarder for Cooja simulator, Serial Line Internet Protocol support for serial forwarder and ultimately a successful TikiriDB application for ContikiOS.

5.0 FUTURE WORK

TikiriDB research has evolved from a simple query based data acquisition solution to a comprehensive information retrieval solution for WSNs. The first version of TikiriDB has been released with basic functionalities and more information about it can be found at http://score.ucsc.lk/projects/tikiridb. During this research new questions and issues emerged and therefore it opened several more other research to improve proposed solution. A certificate revocation mechanism for WSN is one of the areas that haven't been investigated by researchers. In addition to that, one of our groups of researchers at University of Colombo School of Computing (UCSC) is trying to find out a web based solution to access TikiriDB powered WSNs. Then, the solution could be given out for a wider domain of people.

6.0 ACKNOWLEDGMENT

This project is funded by the Computer Systems Laboratory (SICS), Sweden. We appreciate the contributions by Kasun Hewage(UCSC) and Tharindu Nanayakkara(UCSC). We would also like to thank the anonymous reviewers for their valuable comments.

REFERENCES

1: I. F. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, 2002

2: Samuel R. Madden and Michael J. Franklin and Joseph M. Hellerstein and Wei Hong, Tinydb: An acquisitional query processing system for sensor networks, 2005

3: L. Su and B. Ding and Y. Yang and T. Abdelzaher and G. Cao and and J. Hou, oCast: Optimal Multicast Routing Protocol for Wireless Sensor Networks, 2009

4: Adam Dunkels and Björn Grönvall and Thiemo Voigt, Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors, 2004

5: Haodong Wang and Qun Li, Distributed User Access Control in Sensor Networks, 2006

6: Rene Mueller and Gustavo Alonso and Donald Kossmann, D.: SwissQM: Next Generation Data Processing in Sensor Networks,

7: Wai Fu Fung, COUGAR: The Network is the Database, 2002

8: Manabu Isomura and Hiroki Horiuchi and Till Riedel and Christian Decker and Michael Beigl, Sharing sensor networks, 2006

9: Rene Muller and Gustavo Alonso, Efficient Sharing of Sensor Networks,

10: Niki Trigoni and Yong Yao and Alan Demers and Johannes Gehrke, Multi-query optimization for sensor networks, 2005

11: Z. Zhang and A. Kshemkalyani and S. M. Shatz, Multi-Root, Multi-Query Processing in Sensor Networks, 2008

12: Xiang, Shili and Lim, Hock Beng and Tan, Kian-Lee, Impact of multi-query optimization in sensor networks, 2006

13: Qingfeng Huang and Qingfeng Huang and Chenyang Lu and Chenyang Lu and Gruia-catalin Roman and Gruia-catalin Roman, Spatiotemporal Multicast in Sensor Networks, 2003

14: Chen-hsiang Feng and Wendi B. Heinzelman, RBMulticast: Receiver Based Multicast for Wireless Sensor Networks,

APPENDIX

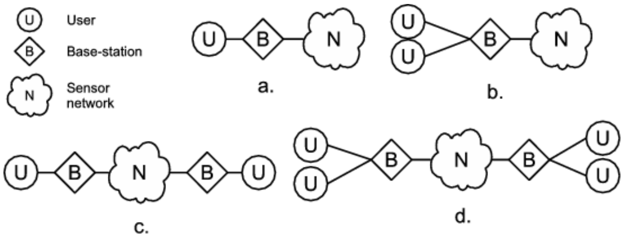


Fig 1: Different scenarios formed by elements of a sensor network

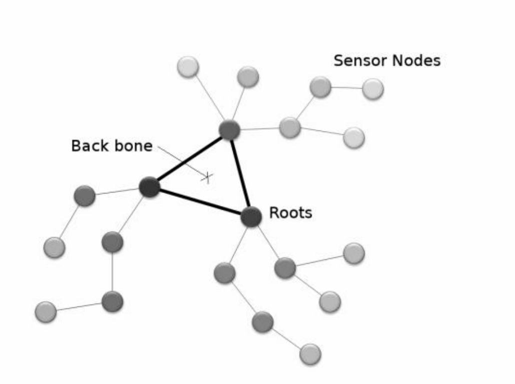


Fig 2: Multi-tree network topology

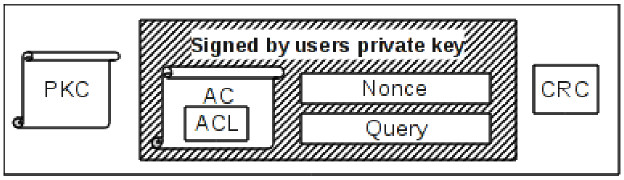
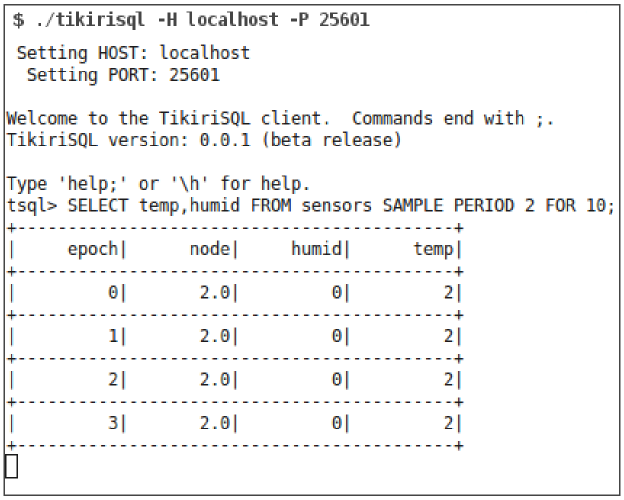


Fig 3: Message from base-station to the

Sensor network



```
$ ./tikirisql -H localhost -P 25601

 Setting HOST: localhost
  Setting PORT: 25601

Welcome to the TikiriSQL client.  Commands end with ;.
TikiriSQL version: 0.0.1 (beta release)

Type 'help;' or '\h' for help.
tsql> SELECT temp,humid FROM sensors SAMPLE PERIOD 2 FOR 10;
+-------------------------------------------+
|     epoch|      node|     humid|      temp|
+-------------------------------------------+
|         0|       2.0|         0|         2|
+-------------------------------------------+
|         1|       2.0|         0|         2|
+-------------------------------------------+
|         2|       2.0|         0|         2|
+-------------------------------------------+
|         3|       2.0|         0|         2|
+-------------------------------------------+
```

Fig 4: Result of the sample query executed in tikirisql text user interface