Query Optimization using Context-Aware Routing Tables for Heterogeneous Sensors

Sang-Won Hwang[1], Young-Kwang Nam[1], Byoung-Dai Lee[2]

[1]Department of Computer Science, Yonsei University, Republic of Korea
irvanz@hanmail.net, ykman@yonsei.ac.kr

[2]Department of Computer Science, Kyonggi University, Republic of Korea
blee@kgu.ac.kr

Abstract: In general, data communication among sensor nodes requires more energy than internal processing or sensing activities. In this paper, we propose a novel technique to reduce the number of packet transmissions necessary for query dissemination or query results relaying processes among neighboring nodes with the help of context-aware routing tables. The important information maintained in the context-aware routing table is which physical properties can be measured by descendent nodes reachable from the current node. Based on the information, the node is able to eliminate unnecessary packet transmission by filtering out the child nodes for query dissemination or query results relaying. The simulation results show that up to 80% of performance gains can be achieved with our technique

[Sang-Won Hwang, Young-Kwang Nam, Byoung-Dai Lee. **Query Optimization using Context-Aware Routing Tables for Heterogeneous Sensors**. *Life Sci J* 2013; 10(3):970-978]. (ISSN: 1097-8135). http://www.lifesciencesite.com. 142

Keywords: metadata; query optimization; routing table; sensor network

1. Introduction

In general, it is not unusal to deploy multi-functional sensor nodes when developing sensor networks. The multi-functional sensor nodes are able to sense different types of physical properties simultaneously such as temperature, humidity, ultrasonic waves, and etc.. Furthermore, they are capable of preprocessing the sensed data internally before transmitting the data to neighboring nodes. In such environments, a wide spectrum of queries must be supported in order to allow the researchers to investigate interesting physcial phenomenon effectively and effiecently.

Regardless of the structures of the sensor networks, the constituent sensor nodes must be able to perform two operations in order to successfully process the queries submitted to the sensor networks: query dissemination and query processing. The query dissemination represents the activities related to delivering queries received from the parent nodes to the child nodes and relaying query results from the child nodes to the parent nodes. The routing tables maintained in the sensor nodes play an important role in finding appropriate parent nodes and child nodes for data transmission. The query processing, on the other hand, involves the activities for sensing the requested physical properties and for preprocessing the data internally such as filtering and aggregation, if necessary, before communicating with neighboring sensor nodes.

Due to the limitation of physical memory size and CPU performance of the sensor nodes, the query optimization in the sensor networks, unlike the query optimization in relational databases that tries to reduce the number of JOIN operations or to minimize the table access time through reordering the computations, focuses on reduction of overal energy consumptions in the sensor nodes [1]. It has been shown that data communication among sensor nodes requires more energy than internal preprocessing or sensing activities [4]. Therefore, in this paper, we propose a novel technique to reduce overall enery consumptions in the sensor nodes by minimizing the number of data communications in the query dissemination process with the help of context-aware routing tables maintained in the sensor nodes. The context-aware routing table of a sensor node keeps track of not only eneygy-efficent packet routing information to the parent and child nodes but also information on the capabilities of all the descendent nodes reachable from the node (Hereafter, we call the information on the capabilities of the decendent nodes as *metadata*). With the routing table, each sensor node is able to determine whether or not it needs to disseminate the query to its child nodes. For example, when all of the descendent nodes reachable from a direct child node *A* of the current node do not have capabilities to measure the temperature, there is no need for the current node to disseminate the query involving the temperature to the node *A*. We call such context-aware routing table as ***MRT (Metadata Routing Table)***. Depnding on which sensor nodes must maintain MRTs, there are two ways in constructing MRTs: ***FMRT (Full Metadata Routing***

Table) and ***SMRT (Sparse Metadata Routing Table)***. With FMRT scheme, all of the constituent nodes in the sensor network maintain MRT, whereas SMRT schme allows only subset of the nodes in the sensor network to maintain MRT in consideration of the cost required to keep the MRTs up-to-date as nodes join/leave to/from the sensor network dynamically over time.

The structure of this paper is as follows. Section 2 shows the previous researches related to the query optimization in the sensor networks and Section 3 describes NanoDB, the query processing system running on Nano-Q+ platform. Section 4 presents in details two MRT techniques employed in NanoDB and Section 5 shows simulation results of the proposed MRT techniques and finally we conclude in Section 6.

2. Related Work

Semantic Routing Tree [4] reduces the amount of packets traversed in the sensor networks by eliminating unnecessary relaying of query results from the childe nodes to the parent nodes. This is done by each node comparing the sensed data from its child nodes against the range of the properties specified in the query and discarding those data residing out of the range. This approach, however, is limited in that the reduction of packet transmissions for query dissemination is not considered.

Dynamic Semantic Routing Tree [1] enhances the Semantic Routing Tree by caching the sensed data from the child nodes for a designated time period and utilizing them for query processing, instead of acquiring the sensed data on-demand from the child nodes. However, due to the use of the cache, the query results may not reflect the up-to-date status of the certain nodes. Furthermore, these approaches are applicable only to the queries involing relational operations.

J. Shneidman *et al* [3] minimizes the communication cost by sharing the results of the aggregation functions among constituent nodes in the homogeneous sensor network.

In E. Ermis *et al* [8], each node exchanges the information of the local sensors so that an event can be revealed through the global performance. This technique characterizes the fundamental trade-offs between the global performance (false alarms and miss rates) and communication costs.

D. Coffin *et al* [10] uses a simple schema to transmit data by collecting information from data-centric protocol rather than address-centric protocol. Rather finding a shortest path to send data to a sink node, a routing schem is pursued by finding a path to nodes where data aggregation function can be applied.

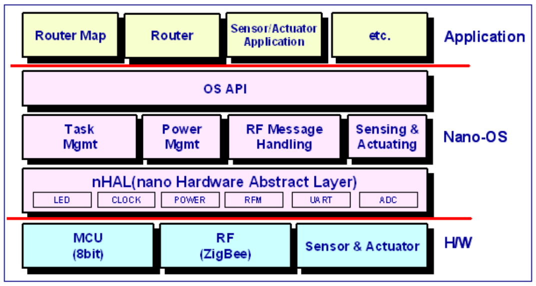

Figure 1. The hierarchical architecture of Nano-Q+ system

Geographic Hash Table [12] hashes keys to geographic coordinates and stores a key-value pair at the sensor node geographically nearest the hash of its key. It uses an efficient consistency protocol to ensure that key-value pairs are stored at the appropriate nodes after topology changes.

In M. Welsh *et al* [14], a sensor network is divided into a set of abstract regions so that a family of spatial operators captures local communication within the regions of the network. This technique exposes the trade-offs between the accuracy and the resource usage for communication operations.

Greedy Perimeter Stateless Routing [15] uses the positions of routers and a packet destination to make packet forwarding decisions. It uses only the information about a router immediate neighbor in the network topology.

Although the abovementioned techniques or algorithms perform efficiently and effectively, none of those deals with heterogeneous sensor networks in which the types of the sensors mounted in the constituent nodes are different.

3. Query Processing System on Nano-Q+ Sensor Network

3.1. Nano-Q+ System

Nano-Q+ system, developed by ETRI (Electronics and Telecommunications Research Institute), is an extensible and reconfigurable embedded system and it consists of Nano-Q+, which is an embedded operating system and Nano-24, which is a sensor hardware running Nano-Q+. Figure 1 shows the hierarchical architecture of Nano-Q+ system. The primary components of Nano-Q+ operating system and their brief descriptions are as follows:

- Nano HAL module – abstracts the underlying snesor hardware. The current implementation of Nano-Q+ provides standard APIs for LED, Clock, power supply, RF module, UART module, ADC (Analog-to-Digital Converter) and Interrupts.

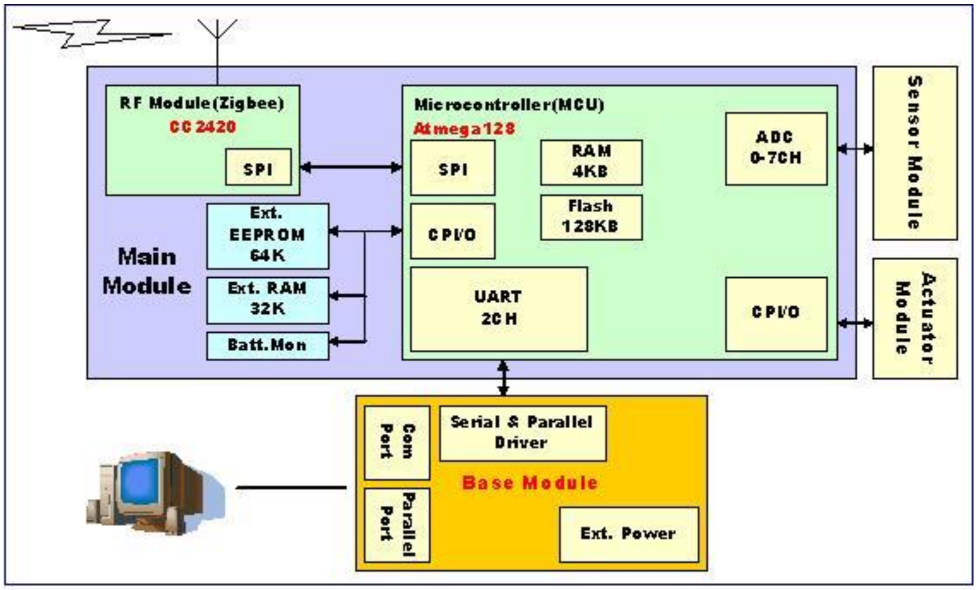

Figure 2. The structure of Nano-24

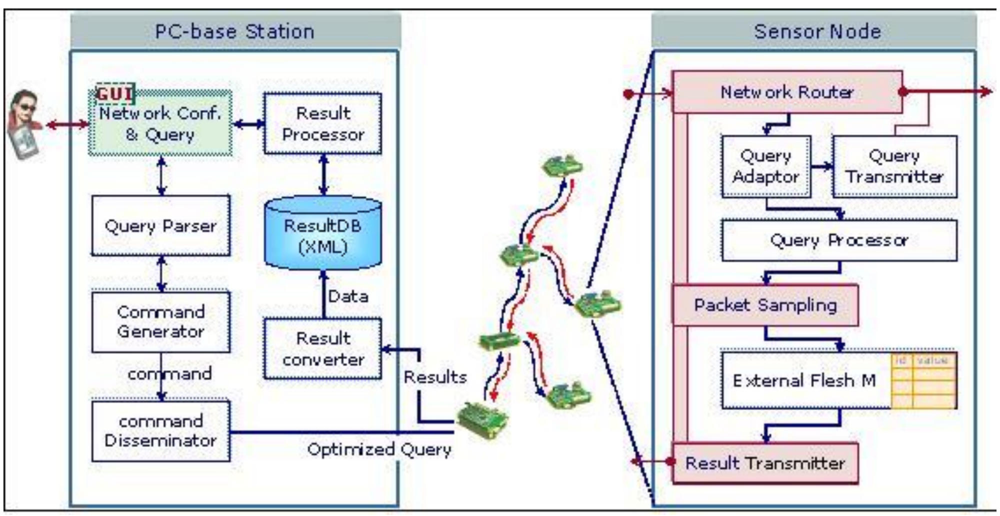

Figure 3. The structure of NanoDB

- Multitask scheduler – is responsible for task management such as task creation/deletion, task scheduling and context switching. The task scheduling is not tied to a specific scheduling algorithm. Instead, the application developers are allowed to choose most efficient algorithm to the applications. Furthermore, the schedule is able to switch to enery efficent mode manually or automatically when there is no task waiting in the ready queue.
- Power management module – monitors the states of MCU (Micro Control Unit) processor and external devices so that it controls the power level dynamically. There are five pre-defined power levels (Idle, ADC Noise Reduction, Power Down, Power Save, Standby).
- RF message handling module – consists of four layers: PHY layer, MAC layer, Link layer and Network layer. The core component of this module is software MAC layer. It implements the requirements of IEEE 802.15.4 in consideration of performance and resource utilization. For example, it miminizes processing time and memory requirements by passing memory locations instead of copying the contents in the memory.

Nano-24 shown in Figure 2 consists of main module, base module, sensor module and actuator module and it is designed in consideration of low cost, low power consumption and high degree of modulation. The core component of the main module is Atmega128L MCU and CC2420 IEEE 802.15.4 RF transceiver. The main module provides 128Kbytes ISR (In-System Reprogrammable) based flash memory, 4Kbytes internal SRAM and 4Kbyte EEPROM. Optionally, it can be equipped with external 512Kbytes flash memory and 32Kbytes SDRM. The base module is composed of RS-232 serial interface, parallel interface, and power management module. Nano-24 is able to sense 7 types of physical properties (temperature, humidity, gas, illumination, ultrasonic waves, superconduction, and velocity) and it operates with two AA batteries.

3.2. NanoDB

NanoDB is a query processing system running on Nano-Q+ sensor network. The system architecture of NanoDB is shown in Figure 3. The base station running on PCs provides two types of interfaces for submitting queries to the sensor network and displaying the final results to users. The graphics-based interface allows users to input range of the physical properties that they are interested and generates the query sytematically whereas with the text-based interface, users are able to compose their own queries with Xquery syntax. The query parser is in charge of validity check of the system-generated or the user-generated queries. Once the query is verified, the command generator translates the query into command packets that will be eventually disseminated to sensor nodes through the command disseminator. The result converter receives query results in the form of packets from sensor nodes and converts them into XML documents and the result processor extracts pieces of information that users are intereated from the XML document and feeds them into GUI component.

When a sensor node receives command packets from its parent node or the base station, it selects a subset of child nodes for query dissemination based on MRT and regenerates new command packets for them. The destination node selection is based on MRT. Once it finishes disseminating the new command packets to the selected child nodes, the sensor node goes into the sleep mode for power saving. On the other hand, when query result from the child node arrives, the sensor node wakes up and keeps the result into the volatile memory. However, if the total size of the partial results from the child nodes exceeds the size of available memory, it stores the results into the external flash moery. When the sensor node recevies all the data from all of the selected child nodes, it filters out the intermediate query results and sends necessary data to its parent node.

4. MRT based Query Optimization

4.1. FMRT

When a sensor node in NanoDB system receives command packets, it does not disseminate the command packets to all of its child nodes. Instead, it selects only a subset of the child nodes based on the capabilities of the child nodes and the descendent nodes reachable from the child nodes. Such information on the capabilities of descendent nodes, which we call medata, is maintained in MRT.

Formally, suppose that $SN = \{S_1, S_2, ..., S_n\}$ represents a sensor network and $SR = \{R_1, R_2, ..., R_m\}$ represents a set of sensor types available in the network. In addition, it is assumed that a set of sensor types available in each paticipant node S_i is different from one another but it is a subset of *SR*.

i) Let $C_{i,t} = \{C^1_{i,t}, C^2_{i,t}, ..., C^p_{i,t}\}$ be a set of child nodes of S_i at time *t* and $C^k_{i,t}$ be a k^{th} child node of S_i.

ii) Let $A_{i,t} = \{B_{i,t}(R_1), B_{i,t}(R_2), ..., B_{i,t}(R_m)\}$ be the capability table of S_i at time *t*, where $B_{i,t}(R_j)$ is set to true if S_i is able to sense R_j.

iii) Let $M^k_{i,t} = \{M^k_{i,t}(R_1), M^k_{i,t}(R_2), ..., M^k_{i,t}(R_m)\}$ be metadata associated with $C^k_{i,t}$ at time *t* and $M^k_{i,t}(R_j)$ is set to true if there exist any descendent nodes (including $C^k_{i,t}$) that a) are reachable from S_i through $C^k_{i,t}$ and b) are able to sense R_j. Otherwise, it is set to false.

iv) Let $M_{i,t} = \{M^1_{i,t}, M^2_{i,t}, ..., M^p_{i,t}\}$ be a set of metadata maintained in S_i, where *p* is the number of child nodes of S_i.

Given a query involving a set of sensor types $QR = \{R'_1, R'_2, ..., R'_r\}$, the node selection process for query dissemination is straightforward. The following shows the algorithm that is executed by node S_i at time *t*:

① Let $T = \phi$
② for each child node $C_{i,t}^{k}$ in $C_{i,t}$, where $k = 1..p$
③ for each $M_{i,t}^{k}(R_j)$ in $M_{i,t}^{k}$, where $j = 1..m$
④ if $(M_{i,t}^{k}(R_j) == true)$ and $(R_j \in QR)$
⑤ $T = T \cup \{C_{i,t}^{k}\}$

If a subtree having a child node $C_{i,t}^{k}$ as a root node is able to sense any of the physical properties that the query is interested, $C_{i,t}^{k}$ is included into the destinations for query dissemination (Line 3-5).

With FMRT scheme, every participant nodes in the sensor network maintain MRTs. Each node receives metadata from the child nodes to update its MRT. Once it is done, the new metadata reflecting the capability information of the current node as well as the descendent nodes must be delivered into its parent node. This process continues until update of MRT of root node is completed because the metadata maintained in a sensor node must reflect the status of all of its descendent nodes. Figure 5 shows the algorithm for MRT updates. Line 1-3 computes the metadata that has been sent to the parent nodes at time *t*. This information, however, can be retained by each node until new metadata needs to be sent to the parent nodes. Note that since

① $V_{i,t} = \{V_{i,t}(R_1) = false, V_{i,t}(R_2) = false, ..., V_{i,t}(R_m) = false\}$
② for each R_k in SR, where $k = 1..m$
③ $V_{i,t}(R_k) = B_{i,t}(R_k) \| (M_{i,t}^{1}(R_k) \| M_{i,t}^{2}(R_k) \| ... \| M_{i,t}^{x}(R_k))$
④ $V_{i,t+1} = \{V_{i,t+1}(R_1) = false, V_{i,t+1}(R_2) = false, ..., V_{i,t+1}(R_m) = false\}$
⑤ for each R_k in SR, where $k = 1..m$
⑥ $V_{i,t+1}(R_k) = B_{i,t+1}(R_k) \| (M_{i,t+1}^{1}(R_k) \| M_{i,t+1}^{2}(R_k) \| ... \| M_{i,t+1}^{y}(R_k))$
⑦ if $V_{i,t} \neq V_{i,t+1}$, send $V_{i,t+1}$ to the parent nodes

Figure 5. Algorithm for MRT updates

① In case that a new node, C_j is added into the network
② if $V_{j,t}(R) = true$
③ remove an entry $M_{i,t}^{j}(R)$ in $M_{i,t}^{j}$
④ send $V_{i,t}(R) = true$ to the parent nodes
⑤ else
⑥ set $M_{i,t}^{j}(R) = false$
⑦ In case that a node, C_j is removed from the network
⑧ if $V_{j,t}(R) = false$
⑨ do nothing
⑩ if $V' = (M_{j,t+1}^{1}(R) \| M_{j,t+1}^{2}(R) \| ... \| M_{j,t+1}^{y}(R)) = false$ and $B_{j,t}(R) = true$, where *y* is the number of child nodes of C_j
⑪ set $M_{i,t}^{j}(R) = false$
⑫ send $V_{i,t}(R) = false$ to the parent nodes
⑬ else if $V_{j,t}(R) = true$ and $B_{j,t}(R) = false$
⑭ do nothing

Figure 6. Algorithm executed by the parent node S_i for MRT updates

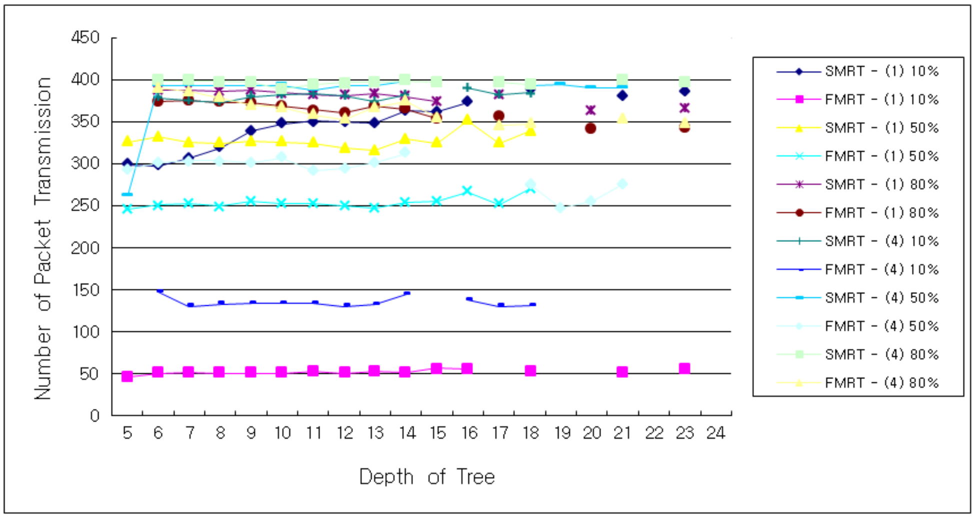

Figure 7. The performance comparison with different types of tree structures

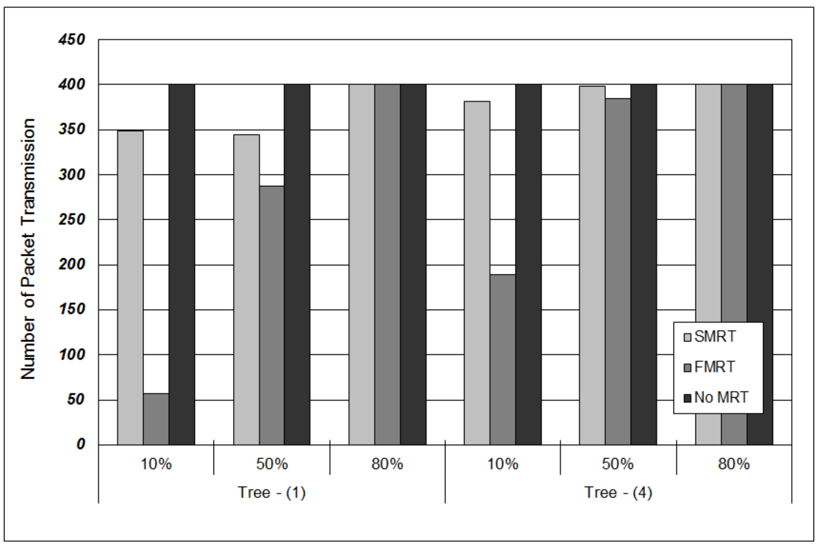

Figure 8. The performance comparison with the sensor types and query satisfaction rate

the metadata delivered to the parent nodes indicates the capabilities of the subtree that the current node resides as a root node, the capabilities of the current node must be considered (Line 3). Line 4-6, on the other hand, computes the new metadata based on the up-to-date information of the capabilities of the descendent nodes as well as the new capability table of the current node. The new metadata is delivered to the parent nodes only when two metadata sets are different, meaning that from the parent node's perspective, the number of the physical properties that a subtree having S_i as a root node is able to measure is increased or decreased. In other words, if the change of node configuration in a subtree does not cause the capabilites of the subtree as a whole, there is no additional matenance cost for keeping MRTs of parent nodes up-to-date.

4.2. SMRT

The important role of MRT is for the current node to identify subtrees containing no sensor types that the given query requires so that the query is not forward to those subtrees, resulting in reduction of packet transmisstions. Although FMRT scheme can achieve this goal, this approach can be inefficient and costly in that the maintenance process of MRTs requires traverse of every node in the network, which accompanies additional computations and communications as well as memory consumptions. In particular, under dynamic environments, where sensor nodes join/leave the network dynamically over time, the cost for keeping the MRTs up-to-date can increase significantly. Therefore, we modified FMRT

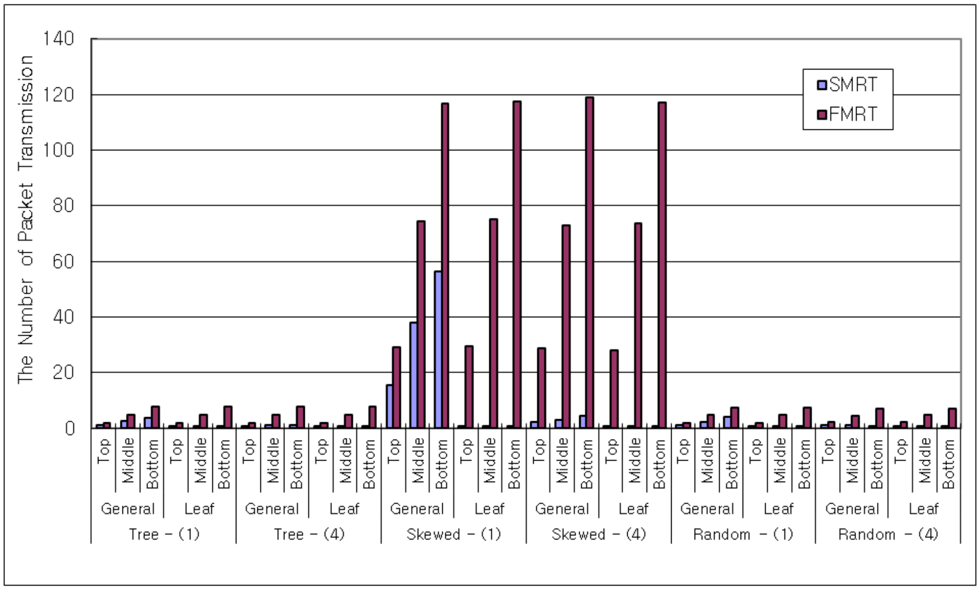

Figure 9. MRT update costs for node addition

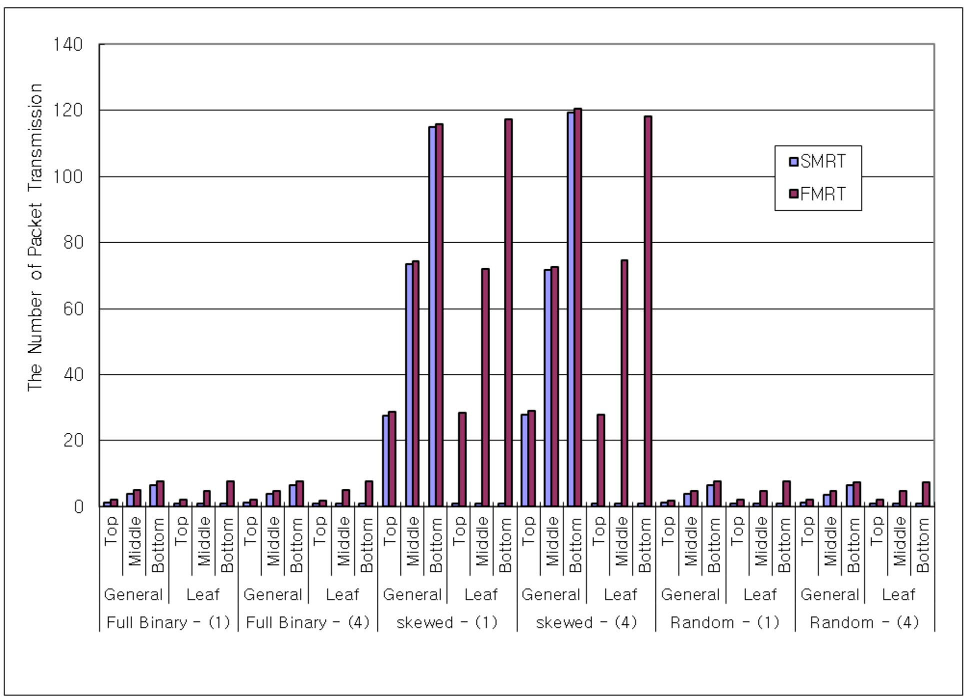

Figure 10. MRT update costs for node removal

scheme to minimize the maintenace cost of MRTs while keeping the pluses of using MRTs, which we call SMRT scheme.

With SMRT scheme, the nodes maintain metadata for decendent nodes only when it is clear that there is no need to forward the queries to the decendent nodes. Furthermore, only the nodes with the ability to sense R_j can maintain the metadata of the descendent nodes for R_j. When all of these conditions are not satisfied, the node removes its MRT and it forwards queries to all of its child nodes. By keeping MRTs dynamically, we minimize the number of the nodes where MRTs must be maintained. Figure 6 shows the algorithm executed by the parent nodes, S_i, for MRT updates. For brevity of explanation, it is assumed that each node has a single sensor of the same type. If the new node and any of its descendent nodes are able to measure R_j , the ancestor nodes remove their MRTs for R_j because the new node must be included to process the queries requiring R_j . This change must be forwarded to all of the ancestor nodes (Line 3-5). On the other hand, if the new node and the subtree having the new node as the root have no ability to measure R_j , then MRTs of the parent nodes must be updated so that the queries are not forwarded (Line 5-6). Note that this update does not affect the MRTs of the ancestor nodes. Similar to node addition, when a node leaves the sensor network, if the node to be removed is the only node to measure R_j , the MRTs of the parent nodes must be updated (Line 10-12). Otherwise, no changes of MRT is needed.

5. Experimental Results

In order to evaluate the performance of MRT schemes, we have built sensor network simulator using JProwler [16]. JProwler, developed by University of Vanderbilt, is an event-driven sensor network simulator that can be set to operate in either deterministic mode (to produce replicable results while testing an application) or in probabilistic mode that simulates the nondeterministic nature of the communication channel and the low-level communication protocol of the sensor nodes [13]. One of the key features of the simulator is that it can evaluate the performances of embedded software from low-level communication layer to application layer by providing simulation models in 4 different areas: Radio propagation models; Signal repetition and collision models; MAC layer models; Application models.

To evaluate different query processing schems, we created a synthetic sensor network consisting of 400 nodes, each of which is able to measure up to 4 different physcial properties. The type of each node is determined randomly. Figure 7 and Figure 8 shows the performances of FMRT and SMRT scheme in randomly generated tree structure as we change the tree depth of the network, the number of sensor types involved in the queries, and the distributions of the target nodes. For example, “FMRT-(4) 80%” represents the performance of FMRT scheme for processing a query involving 4 sensor types and 80% of the nodes satisfy the query. As the number of involved sensor types and the percentage of target nodes decrease, FMRT shows better performance than SMRT. However, As the percentage of target nodes increases, the performance of SMRT approaches that of FMRT. Interestingly, with “SMRT-(1) 10%” and “SMRT-(1) 50%”, the performance descreases as the depth of the tree increases even though the percenages of target nodes decreases. This is because SMRT allows only subset of nodes to maintain MRTs and the location of MRTs in the tree affects the overall performance. For example, if the nodes with MRTs are located on higher levels of the tree, there is higher chance that unnecessary query dissemination is filtered out.

Figure 9 and Figure 10 show the number of packet transmissions required to update MRTs in different network configurations such as standard trees, skewed trees and randomly generated trees. We also measured the effects of the locations of the nodes to be added/removed. The results clearly show that SMRT scheme requires less maintenace cost than FMRT. This is mainly because when there is change in network configuration, SMRT requires the necessary data to be forwarded to only those nodes in which MRTs exist. However, with FMRT, the update must be traversed to the root node.

6. Conclusions

In this paper, we presented a novel technique for query optimization that utilize context-aware routing table, which we call MRT. The MRT maintains capabilities information of subtrees having the child nodes as root nodes. With MRTs, nodes are able to reduce the number of packet trasmissions required for query dissemination by discarding the child node if all of the descendent nodes of the child node as well as the child node cannot measure the physical properties specified in the query.

The experimental results show that FMRT scheme, where all of the constituent nodes maintain MRTs, achieved the best performance but the mainteance cost for MRT is significantly higher than SMRT scheme in most cases. The performance of SMRT scheme approaches that of FMRT scheme as the percentage of the target nodes increases. According to the experiments, when the percentage of the target nodes resides between 50% and 80%, the best performance can be achieved. In addition, it is recognized that the location of nodes with MRTs can affect the overall performance of SMR scheme along with the distribution of the target nodes.

The future work is to enhance SMRT scheme in order to reduce the effects of the location of the nodes with MRT. In particular, we are investigating the possibility of using FMRT and SMRT simultaneously.

References

[1] J. Zhu, “Dynamic Semantic Routing in Sensor Network,” Technical Report, Dept. of CS, Rochester Institute of Technology, NY, USA,

2004.
[2] J. Gehrke and S. Madden, "Query Processing in Sensor Network," *IEEE Pervasive Computing*, vol. 3, pp. 46-55, 2004.
[3] J. Shneidman, P. Pietzuch, M. Welsh, M. Seltzer, and M. Roussopoulos, "A Cost-Space Approach to Distributed Query Optimization in Stream Based Overlays," *Proc. of IEEE Int'l Conf. on Networking Meets Databases*, 2005.
[4] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TinyDB: An Acquisitional Query Processing System for Sensor Networks," *ACM Transaction on Database System*, vol. 30, no. 1, pp. 122-173, 2005.
[5] Y. Yao and J. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," *ACM SIGMOD Record*, vol. 31, no. 3, pp. 9-18, 2002.
[6] A. Agarwl, and T.D.C. Little, "Attribute Based Routing in Sensor Networks," Technical Report, Boston University, 2006.
[7] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajarman, "Multi-Query Optimization for Sensor Networks," *Proc. of the 1st IEEE Int'l Conf. on Distributed Computing in Sensor Systems*, pp. 307-321, 2005.
[8] E.B. Ermis and V. Saligrama, "A Statistical Sampling Methods for Decentralized Estimation and Detection of Localized Phenomena," *Proc. of 4th Int'l Symp. on Information Processing in Sensor Networks*, 2005.
[9] B. Krishnamachari, D. Estrin, and S. Wicher, "Modeling Data-centric Routing in Wireless Sensor Networks," *Proc. of IEEE Infocom*, 2002.
[10] D. Coffin, D.V. Hook, S. McGarry, and S. Kolek, "Declarative ad-hoc Sensor Networking," *SPIE Integrated command Environments*, 2000.
[11] R. Govindan, "Data-centric Routing and Storage in Sensor Networks," Wireless Sensor Network, Kluwer Academic Publishers, pp. 185-205, 2004.
[12] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: Geographic Hash Table for Data-centric Storage," *Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*, pp. 78-87, 2002.
[13] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation-based Optimization of Communication Protocols for Large-scale Wireless Sensor Networks," *Proc. of IEEE Aerospace Conference*, pp. 1339-1346, 2003.
[14] M. Welsh and G. Mainland, "Programming Sensor Networks Using Abstract Regions," *Proc. of the 1st Conf. on Networked Systems Design and Implementation*, pp. 3-3, 2004.
[15] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. of 6th Annual Int'l Conf. on Mobile Computing and Networking*, 2000
[16] http://www.isis.vanderbilt.edu/Projects/nest/jprowler/index.html.

4/12/2013