http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (10): 1934-1941, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1934.1941

An Efficient Index-based Data Storage Method for Wireless Sensor Networks

Chao Gao, Bingwen Wang, Xiaoya Hu, Zhuguo Li and Hongliang Gao
Department of Control Science and Engineering,
Huazhong University of Science and Technology, Wuhan Hubei, 430074, China

Abstract: In Wireless Sensor Networks (WSNs), various schemes have been proposed to efficiently store and process sensed data. The appropriate method to store sensed data depends on the applications. In this study, a data storge scheme called Distributed Index-based Dominating Set (DIDS) was proposed to reduce the communication cost for transmitting data and to efficiently process data queries. This scheme is based on the idea that sensing data are stored at the nodes closed to the detecting nodes and the location information of these storing nodes is pushed to some index nodes closed to them.So queries are only needed to be routed to the index nodes instead of flooding into the whole network. Therefore, the DIDS scheme can minimize the use of limited network and computational resources while providing timely responses to queries. Moreover, This data dissemination scheme also disscusses load balance. Analysis and simulations show that the DIDS scheme outperforms the External Storage (ES) based scheme, Local Storage (LS) based scheme and the Data Centric Storage (DCS) based scheme.

Key words: Wireless sensor networks, data storage method, distribute index

INTRODUCTION

Wireless Sensor Networks (WSNs) provide a new opportunity for pervasive and context-aware monitoring of physical environments. A WSN is composed of numerous sensor nodes, each being a tiny wireless device that can continuously collect environment information and report to a remote sink through a multi-hop network (Akyildiz *et al*., 2002). A WSN is usually deployed in a region of interest to observe particular phenomena or track objects inside the region. Practical applications of WSNs include, for example, habitat monitoring (Tovar *et al*., 2010; Chu *et al*., 2011), health care (Poon *et al*., 2006), smart home (Helal *et al*., 2005) and parking systems (Idris *et al*., 2009a, b). As we know, energy and storage capacity are two kinds of the most important resources in wireless sensor networks.

There are several previous works for data storage and query processing in wireless sensor networks. One widely adopted scheme is the External Storage-based (ES) data dissemination (Ratnasamy *et al*., 2003). It relies on a centralized base station which is external to the sensor network, for collecting and storing sensing data. If data updates and queries very frequently from nodes within the network, this scheme is very inefficient since data must be sent back and forth between the sensors and the base station. To avoid unnecessarily transferring the sensing data, Local Storage-based (LS) data dissemination schemes, e.g., directed diffusion (Intanagonwiwat *et al*., 2003), have been proposed. In these schemes, a source sends data to a sink only when the sink has sent a query for the data. These schemes need a sink source matching mechanism to facilitate a sink to find the source holding the data of interest. The matching mechanisms adopted by most LS schemes follow a flood-response pattern which inherently needs to flood certain control messages. For example, in directed diffusion, a sink floods its query over the whole network; the source(s) with the requested data then knows where to send the data. Considering a large scale WSN, the network-wide flooding may introduce significant traffic in the Local Storage-based (LS) scheme. In Data Centric Storage (DCS) (Ratnasamy *et al*., 2003), the sensing data of an event (e.g., antelope sightings) are stored at certain nodes within the network. In this scheme, however, data are still pushed in a predefined manner regardless of queries. Hence, it lacks flexibility and may introduce a huge unnecessary communication overhead when the querying rate is low, especially for large scale WSNs.

In order to provide low average query and seek to balance communication requirements over participating nodes, He *et al*. (2005), Zhang *et al*. (2007), Yiwei and Yingshu (2009) and Ruan *et al*. (2010) investigate constructing index nodes to facilitate query processing. It seems that any kind of index on distributed data requires a hierarchical structure to aggregate information

Corresponding Author: Chao Gao, Department of Control Science and Engineering,
Huazhong University of Science and Technology, Wuhan Hubei, 430074, China

from the whole network. More detailed information can be accessed by a top-down traversal of the hierarchy to visit the sensors holding the relevant information. A number of methods have been presented to construct such hierarchies for query processing and data dissemination. However, most of those works adopt a tree-based index which is widely used in traditional databases. Since there does not exist any existing or pre-defined network infrastructure, it is difficult to maintain this tree-based index structure in WSNs. Hence, we adopt another important structure called k-hop Connected Dominating Set to form storage and index node sets.

The CDS-based scheme is different from the hierarchical structure based on clustering protocol (Heinzelman *et al*., 2002; Dai *et al*., 2009; Wang *et al*., 2009, 2011). A Connected Dominating Set (CDS) working as a virtual backbone is an effective way to decrease the overhead of routing in a wireless sensor network (Liu *et al*., 2010a, b). For a graph G (V, E), where V is the node set and E is the edge set, a Dominating Set S of G is defined as a subset of V such that each node in V/S is adjacent to at least one node in S. A Connected Dominating Set (CDS) C of G is a dominating set of G which induces a connected subgraph of G. Figure 1a shows an example of CDS. The nodes in C are called dominators, the others are called dominatees. A k-hop dominating set D in G is a set of nodes with the property that every node in G is at most k hops away from at least one of the nodes of D. Figure 1b shows an example of 2-hop dominating set. A CDS is the earliest structure proposed as a candidate for virtual backbones in WSNs. Not only a CDS can be used to facilitate routing, it also can be used as data dissemination framework.

In this study, a novel distributed index based k-hop DS data dissemination and storage scheme is proposed. In the scheme, the sensing data are stored at the storage nodes close to the sensing node. A storage node only sends data to a sink when it receives a query from the sink. Also, the location information (called index) of the storage nodes are pushed to and maintained at some nodes (called index nodes) some hops away them. Hence, queries are routed to the appropriate index nodes. This scheme is more attractive than the existing data dissemination schemes since it avoids both unnecessarily transferring the sensing data and flooding control messages to the whole network. Certainly, this scheme introduces additional overhead for maintaining index nodes.

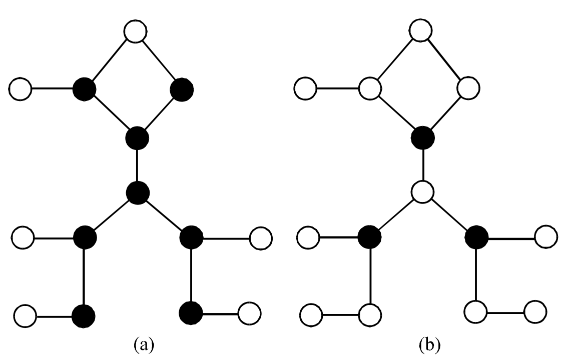

Fig. 1 (a-b): (a) is an example of CDS and (b) is an example of 2-hop DS, where black nodes represent dominators

RELATED WORK

Construct a distributed index for a WSN to facilitate data dissemination has received some attention only recently, even though the index has been extensively studied in traditional database systems.

The study by He *et al*. (2005) addresses the time-index problem. The main idea is to find disjoint CDS for each time slot and then use the dominator nodes in the CDSs as storage nodes where queries are flooded through all CDS nodes. This scheme mixes the storage nodes and index nodes into one layer and does not consider index independently. It cannot balance the query and storage communication over participating nodes.

An integrated distributed CBI data dissemination scheme was proposed to facilitate data dissemination in large scale WSNs (Yiwei and Yingshu, 2009). It uses the method of connected dominating set but it is not taken into account the issue of energy in the topology construction and maintenance of the node. As we know, in this method storage nodes are overload, if energy is depleted, the data stored in the storage node will be lost.

The main contribution of this study is that we propose a new data dissemination scheme called Distributed Index-based Dominating Set (DIDS) scheme. The DIDS can support scalable handling of large amount of sensing data to overcome the drawbacks mentioned above. In this scheme, sensing data is collected and stored at the storage nodes which form an s-hop dominating set of the whole network. Meanwhile, the information of high level semantically rich data is pushed and maintained at some index nodes formed by a connected i-hop dominating set. Hence, queries are only needed to be routed to the appropriate index nodes instead of flooding into the whole network. The DIDS data dissemination scheme can minimize the use of limited network and computational resources while providing timely responses to queries and ensure scalability and load balancing of communication as well as adaptivity in the presence of dynamic changes.

DIDS SCHEME AND ANALYSIS

The basic idea of DIDS scheme: In this study, static symmetric multi-hop wireless networks are discussed. The topology of a wireless sensor network is represented by a graph G = (V, E), where V is the node set and E is the edge set. If two nodes are within the transmission range of each other, then there is an edge between them. We assume that all nodes are deployed in a 2-D plane, the area covered by a node is a circle with the radius equals this node's sensing range. There are many targets moving within a vast region. The sensor nodes in the network can detect the status of each target in its sensing range and periodically generate sensing data. Users may issue a query via sink for data about the current activities of each target. The result report also will be returned to the user via the sink.

In DIDS scheme, the network is logically divided into three layers as Fig. 2 shows. The bottom layer contains the sensing nodes that monitor the targets and generate raw sensing data. This layer also has some special sensing node called standby node that may become storage node later.

The middle layer contains the storage nodes that are used to store the data. We construct an s-hop dominating set of the whole network and use it as a storage node set. The basic idea of how to choose storage nodes is that sensing data are collected and stored at the nodes close to the sensing nodes. Consequently, the maximum distance between the sensing nodes and storage nodes is at most s hops. Therefore, the raw data do not need to travel across the whole network. Thus, it can save limited network resource.

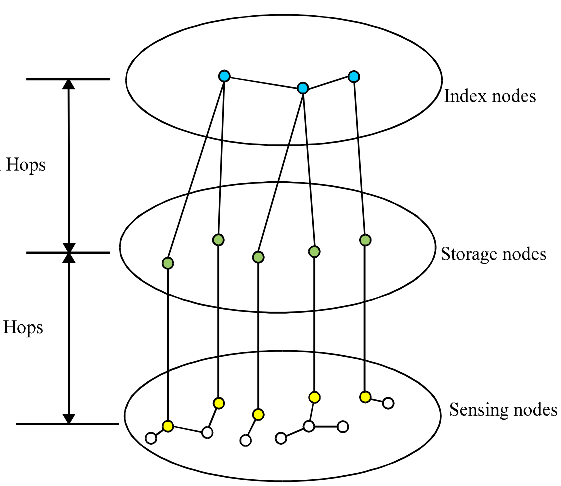

Fig. 2: Hierarchical structure of DIDS scheme

The top layer contains the index nodes that stored high-level semantic data called index information. The main design goal of a distributed index scheme is to provide timely and efficient response to queries while minimizing the amount of network and computational resources consumed by data dissemination. Hence, a connected i-hop dominating set is used as index node set to dominate the storage nodes only.

According to the property of this connected i-hop dominating set, the maximum distance between storage nodes and the index nodes is at most i hops and the index node set is strictly connected as well.

Obviously, storage nodes and index nodes in general consume more energy in handling various bypass traffic than sensing nodes. Those bypasses overhead will drain the energy of those storage nodes and index nodes very quickly, so storage node is the bottleneck of the system. Because the sensing node only collected data, its data are transferred to the storage node for storage. If the number of children nodes is large, data storage and transmission, data query will consume a lot of energy and storage capacity of storage node. The DIDS scheme use two methods to solve prolong the life span of the whole network. Firstly, index nodes do not need to do sensing task to save energy. Secondly, each storage node is equipped with a standby node to reduce overload problem.

Wherever a query is injected into the network via a sink, it would follow a path from the storage nodes to the index node which stores the index information. Thus, the access time from the sink to the index node is a constant which is at most s+i hops. That means the DIDS scheme can provide timely and efficient response to queries.

STORAGE NODES DETERMINATION

According to the definition of s-hop dominating set, every node in G is at most s hops away from at least one of the nodes in an s-hop dominating set of G. Hence, the DIDS construct an s-hop Dominating Set (DS) of the whole network and all the nodes in this set are used as storage nodes. In this way, sensing data are collected and stored at the nodes close to the sensing nodes. Since sensor nodes might generate a significant amount of redundant data, another advantage of using an s-hop dominating set as a storage node set is that storage nodes can combine data by using data suppression or data aggregation (Lindsey *et al*., 2002; Kai-Wei Fan *et al*., 2007).

We firstly introduce a WEIGHT FUNCTION which is W (Level, Energy, Degree, ID), where Level is the hop distance from the root after the BFS search, Energy is the ratio of residual energy to total energy, Degree is the number of the neighbors in the network and ID is the node's unique ID, each node has a unique ID in DIDS scheme.

Theorem 1: The weight function W (L_1, E_1, D_1, ID_1)>W (L_2, E_2, D_2, ID_2), if any of the following conditions is true:

- $L_1<L_2$,
- $L_1<L_2$, $E_1>E_2$,
- $L_1<L_2$, $E_1>E_2$, $D_1>D_2$,
- $L_1<L_2$, $E_1>E_2$, $D_1>D_2$, $ID_1>ID_2$

Algorithm 1: Storage node dermination (G (V, E)

1 : Choose a root r ∊ V(i.e., useing a leader selection alogorithm)
2 : Compute the hop distance from r to each node (useing breadth first search)
3 : Every node exchange information with its nods
4 : Color every node in V white
5 : While a white node in V exits do
6 : The white node u with the smalllest W send a DOMINTING message to its s-hop away parent v having the largest W
7 : v bocomes green, broadcasters a GREEN message to its s-hop neighbors
8 : If receive a GREEN message, white node become gray
9 : end while
10 : Add all green nodes into C
11 : Return C

Algorithm 1 shows the details of the storage nodes determination. To obtain the storage nodes set, the DIDS scheme chooses a root to start a Breadth First Search (BFS). The root can be selected by using leader election techniques. Then the root initializes a BFS search. After that, every node exchanges information with its s-hop neighbors about the level, energy, degree and ID. All nodes should know all its s-hop neighbors' information as a result. If a node has no children node, it is called a LEAF node.

Secondly, the leaf node u with the smallest W (Level, Energy, Degree, ID) sends a DOMINATING message to its exact s-hop away parent v having the largest W (Level, Energy, Degree, ID) to request v to become a dominator (storage node). Whenever v receives this DOMINATING message, it becomes green and broadcasts a GREEN message to all of its s-hop neighbors. Upon receiving a GREEN message from its parent, u becomes gray and broadcasts a GRAY message to all of its s-hop neighbors. Then, the next node with the smallest W (Level, Energy, Degree, ID) among its s-hop neighbors that have not decided their status yet starts this procedure in turn. It becomes gray and broadcasts a GRAY message to its s-hop neighbors if it has received at least one GREEN message. Otherwise, it sends a DOMINATING message to its exact s-hop away parent with the largest W

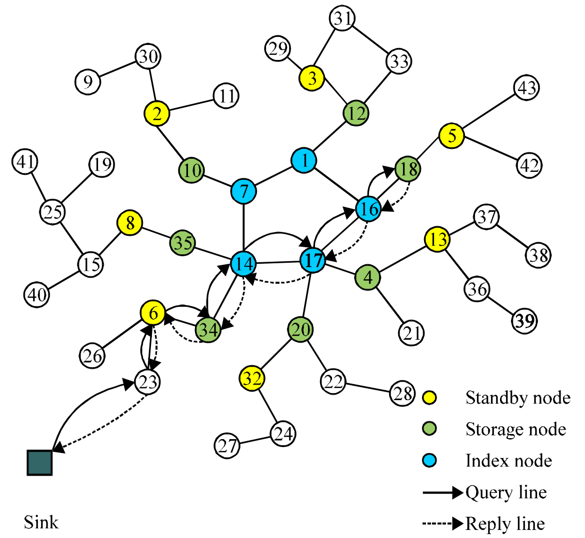

Fig. 3: An example of storage nodes, standby nodes and index nodes determination where s = 3 and i = 1

(Level, Energy, Degree, ID) as u does. This procedure stops when the root becomes gray or green. After that, all green nodes are used as storage nodes that can s-hop dominate the whole graph as well. As shown in Fig. 3.

INDEX NODES DETERMINATION

Algorithm 2: Index node determination (G' (V', E'))

1 : Let G'(V', E') donote the subgraph induced by the node in C and their parents in V
2 : if r∊C
3 : Choose a new root r∊V' (choose its white neighbor with the largest W)
4 : Compute the hop distance from r to each node (useing breadth first search)
5 : Every node exchange information with its nodes
6 : else
7 : Color every node in V' white
8 : While a white node in C exits do
9 : The white node u in C with the smallest W send a DOMINTING message to its i-hops away parent v having the largest W
10 : v becomesblue, broadcasts a BLUE message to its I-hops neighbors
11 : if receive a BLUE message, white node become gray
12 : end while
13 : Add all blue nodes into D
14 : return D

As Algorithm 2 shows, it construct a connected i-hop dominating set I to dominate all storage nodes only and use all the nodes in I as index nodes. Here, strict connectivity is required. Firstly, the algorithm form a new graph G' which includes all the storage nodes and their parents in G. Then it constructs a connected i-hop dominating set to dominate storage nodes only. However, all the storage nodes should be dominatees since they should not be index nodes. Therefore, if the root becomes a storage node, it should pick up another node as the new one to start a new BFS search. The easiest way to find a new root is to transfer the role from the root to its white neighbor with the largest W (Level, Energy, Degree, ID), denoted as r'. From r' it builds a BFS tree but restrict in G' only and also keep the storage nodes as leaf nodes as well.

After that we can use the same procedure used in the storage node determination to find the dominators firstly. However, since the scheme only need to dominate all the storage nodes, only a storage node should send a DOMINATING message to its exact i-hop away parent with the largest W (Level, Energy, Degree, ID).

Once find all the dominators and color them blue, we should add some connectors to connect all those dominators. In here, each blue node finds the shortest path to the root and marks all the internal nodes blue.

Finally, all the blue nodes form a connected i-hop dominating set which dominates all the storage nodes only. All the nodes in the set D are used as index nodes.

Standby nodes determination: Because the storage nodes consume more energy in handling various bypass traffic, they may run out energy soon. So each storage node is equipped with a standby node in DIDS scheme. We select the standby node in the storage node's 1-hop children nodes. If more than one, select the node with the largest (Level, Energy, Degree, ID), as shown in Fig. 3.

The DIDS set a power threshold Ke, whose value is between the 0~1. When the sensor node's residual energy K<Ke, storage node sends the control packet to its index node as well as its standby node, to activate standby node, making it a new storage node. As a result, all new collected data will be stored in the new storage node, the former original storage node becomes a common sensing node that provides the necessary historical data query and data forwarding as well as sensing data. This method can maintain the stability of WSN to reduce the network reconfiguration control packet overhead, the storage structure of the system only to generate 1-hop change.

For example, as shown in Fig. 3, Node 6 is the standby node of the storage Node 34. If the sensor Node 34's residual energy ratio K<Ke, Node 34 sends the control packet to its index Node 14 as well as its standby Node 6 .Then Node 6 will be the new storage node in this subtree. Node 23, 26 and 34 and itself will store data in the new storage Node 6. Index Node 14 will be Node 6's index node; Node 34 will be a sensing node.

COMMUNICATION AND TRAFFIC COST

Here, the overhead of the performance of DIDS with other data dissemination schemes, ES, LS and DCS is compared. To compare the overhead of different data dissemination schemes, we introduce the following notations:

Notations:

s	:	The max hops from sensing node to storage node
i	:	The max hops from storage node to index node
N	:	The total number of nodes in the network
n	:	The total number of targets
r_u	:	The rate of updating of events
r_q	:	The rate of event query
s_d	:	The size of a data report
s_q	:	The size of a query message
s_i	:	The size of an index update message

Definitions:

- **Total message complexity:** The number of messages generated in the whole network
- **Hotspot message complexity:** The maximum number of messages sent, received and forwarded by one single node in the network
- **Total traffic complexity:** The amount of data transferred in the whole network
- **Hotspot traffic complexity:** The maximum amount of data sent, received and forwarded by one single node in the network

There are two kinds of query results which will be returned to the users. The first one is a summary of the events. Thus, the query only need to be processed at the index nodes and the result report will be generated by those index nodes. The other one is a list of each event satisfying the query conditions. In this case, not only the query will be processed in the index nodes, the storage nodes that hold the corresponding data also need to generate the result report and forward it to the sink. In order to simplify communication cost analysis, it only considers the second one where users care about a list of the events rather than a summary of each one. For example, one might want a list of each target sighting rather than just a count of the number of targets seen. In most cases there exist some targets in the monitored area, only one target's activity will be returned per query in this analysis.

Table 1 lists the approximate costs of communication and traffic complexities. The tables can be clearly seen that the average network lifetime is decided by the total and hotspot traffic complexities.

Observation 1: The DIDS scheme has smaller overall message complexity than the other schemes, if:

Table 1: Estimating the overhead of the data dissemination schemes

Schemes	Total messages complexity
(a)Total communication costs	
ES	$nr_u\sqrt{N}$
LS	$r_q(N+n\sqrt{N})$
DCS	$nr_u\sqrt{N}+2r_q\sqrt{N}$
DIDS	$n(i+s)r_u+3r_q\sqrt{N}$
Schemes	Hotspot messages complexity
(b) Hotspot communication costs	
ES	nr_u
LS	$(n+1)r_q$
DCS	nr_u+2r_q
DIDS	nr_u+3r_q
Schemes	Total traffic complexity
(c) Total traffic costs	
ES	$nr_us_d\sqrt{N}$
LS	$r_qs_qN+nr_qs_d\sqrt{N}$
DCS	$(nr_us_d+r_qs_q+r_qs_d)\sqrt{N}$
DIDS	$(ss_d+is_i)nr_u+(s+i+\sqrt{N})r_qs_q+2r_qs_d\sqrt{N}$
Schemes	Hotspot traffic complexity
(d) Hotspot traffic costs	
ES	nr_us_d
LS	$r_qs_q+nr_qs_d$
DCS	$nr_us_d/2+r_q(s_q+s_d)$
DIDS	$r_u(s_d+s_i)+r_q(s_q+s_d)$

- N is large enough
- $r_u>>r_q$ which can be derived by comparing Table 1a

Observation 2: The hotspot message complexity of DIDS is still as good as DCS, if $r_u>>r_q$ which can be derived by comparing Table 1b.

Observation 3: As we know, in most cases, $s_d>>s_q$ and $s_d>>s_i$. The DIDS scheme has smaller Hotspot traffic complexity than the other schemes, if:

$$r_q/r_u>2/(n-1)$$

which can be derived by comparing Table 1c and d.

PERFORMANCE EVALUATION

Here, we conducted simulations to evaluate our scheme.

As we know, a data packet is much larger than an index packet. Therefore, it is desirable that i>s, so we choose s = 3 and i = 4 and the total number of nodes is 1000 in this simulation. Other parameters are listed as Table 2. The size of a query message s_q is 10 while the size of a data message s_d is 70. The size of an index update message s_i is 10. Ten mobile targets are randomly generated in the network to simulate objects moving in the area, each target randomly picks up a direction to move. It returns along the previous path whenever it moves out the boundary of the monitored area. A sensor node can detect the target whenever the target is in its sensing range. We also assume that the result of one target is returned for each query. The simulation duration time is 200.

Table 2: The Simulation parameters

Symbol	Definition	Setting
s	The max hops from sensing node to storage node	3
i	The max hops from storage node to index node	4
N	The total number of nodes in the network	1000
n	The total number of targets	10
s_d	The size of a data report	70
s_q	The size of a query message	10
r_u	The rate of updating of events	0.7
s_i	The size of an index update message	10

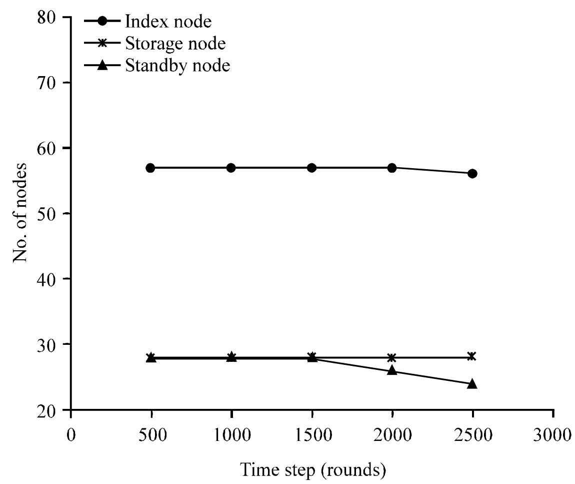

Fig. 4: The size of index node set, storage node set and standby node set

Figure 4 plots the size of storage node set, standby node and index node sets with time step growth by using our constructing method. The size of standby node set decreases with the increasing of time step as shown in Fig. 4. However, when time step increase, the size of storage node set unchanged since some standby node can change the role from standby node to storage node. The size of storage node set, standby node set and index node set is much smaller than the total number of nodes in the whole network. This observation shows that DIDS scheme has good performance in logically classifying the whole network to different layers.

Figure 5 shows the message and traffic complexities Comparison of LS, ES, DCS and DIDS schemes. The average network lifetime is decided by the total and hotspot traffic complexities obviously. Overall, we can see that no single scheme outperforms the others in all cases from the results.

As shown in Fig. 5a, the total message complexity of DIDS decreases significantly when r_u/r_q is larger than 1.5

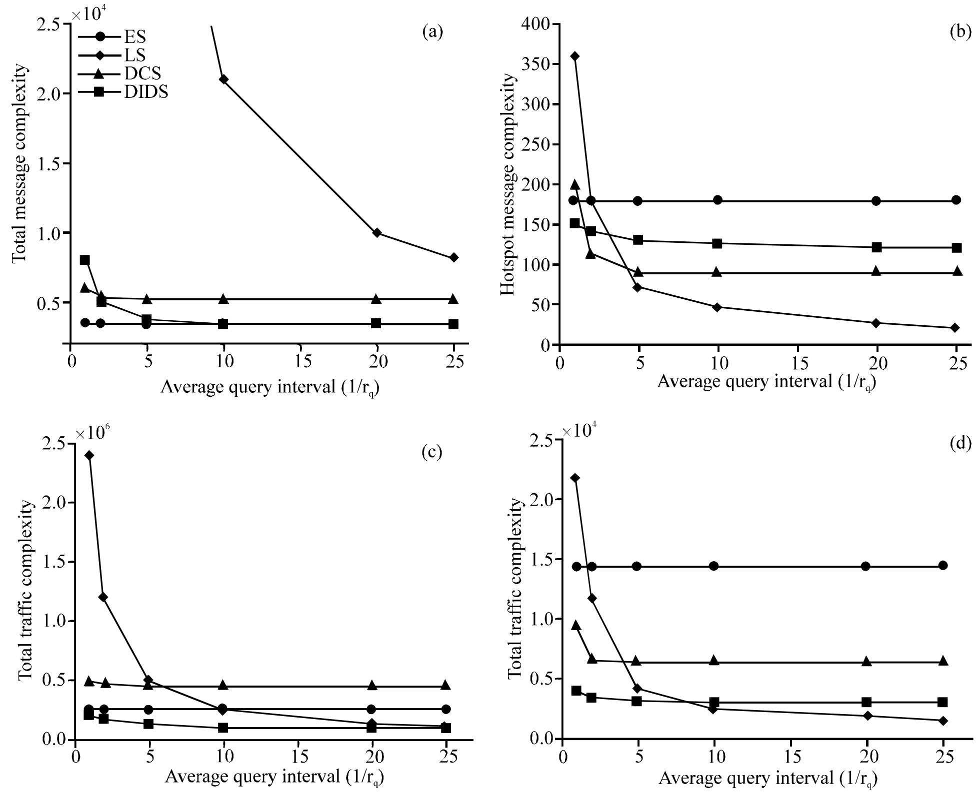

Fig. 5 (a-d): The comparison of ES, LS, DCS and DIDS, (a) Total message complexity, (b) Hotspot message complexity, (c) Total traffic complexity and (d) Hotspot traffic complexity

while it is a little bit larger than LS and DCS when r_u/r_q is less than 1.5. This means DIDS has good performance when query ratio is low, the reason is the data packet or a query does not need to be forwarded across the whole network as others do. This means that DIDS can save valuable energy resources as well as reduce network congestion.

When r_u/r_q is large than 2, the hotspot message complexity of DIDS is larger than LS and DCS, as shown in Fig. 5b. However, The DIDS is still very competitive with LS and DCS. Moreover, DIDS has good performance in hotspot message complexity when r_u/r_q is smaller than 2. Because the queries are restricted to the index nodes, the number of which is smaller enough compared with the total number of nodes in the whole network.

As shown in Fig. 5c, the total traffic of DIDS is much lower than others because of the property of DIDS. In DIDS scheme, sensing data are collected and stored at the nodes close to the sensing nodes and queries are restricted in a small number of nodes.

The hotspot traffic complexity of DIDS is larger than LS when r_u/r_q is larger than 8 as shown in Fig. 5d, since the query ratio is a dominating factor in DIDS and LS. In fact, there is only a little difference between DIDS and LS and the DIDS is still better than ES and DCS.

CONCLUSION AND DISCUSSION

In this study, we propose a Distributed Index-based Dominating Set (DIDS) data dissemination and storage method to support scalable handling of large amount of sensing data in wireless sensor networks. In this scheme, sensing data is collected and stored at the storage nodes which form an s-hop dominating set of the whole network. Meanwhile, the information of high level semantically rich data is pushed and maintained at some index nodes i-hop away from the storage node. Whenever a query is injected into the network via a sink, it would follow a path from the storage nodes to the index node which stores the index information. Thus, the access time from the sink to the index node is a constant which is at most s+i hops. That means the DIDS scheme can provide timely and efficient response to queries while minimizing the use of limited network and computational resources. The DIDS scheme also ensures scalability and load balancing of communication as well as adaptivity in presence of dynamic changes. From analysis and simulation results, we can conclude that the DIDS scheme has better performance than LS, ES, DCS in message and traffic complexities especially for r_u/r_q is no more than 8.

Our future work is to investigate an efficient index data structure which can inherit the index structure. Another direction of future work is how to maintain our data dissemination scheme in presence of network dynamic changes. In addition, we are planning to develop a testbed using Berkeley Motes to validate the proposed schemes.

ACKNOWLEDGMENTS

This study is supported in part by a grant from the National Science Foundation of China (No. 6080202).

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. A survey on sensor networks. IEEE Commun. Magazine, 40: 102-114.

Chu, H., Z. Xie and H. Jiang, 2011. Hybrid intelligent monitoring network based on Ad hoc and wireless sensor networks. Inform. Technol. J., 10: 849-855.

Dai, Z., Z. Li, B. Wang and Q. Tang, 2009. An energy-aware cluster-based routing protocol for wireless sensor and actor network. Inform. Technol. J., 8: 1044-1048.

He, G., R. Zheng, I. Gupta and L. Sha, 2005. A framework for time indexing in sensor networks. Acm Transact. Sensor Networks, 1: 101-133.

Heinzelman, W.B., A.P. Chandrakasan and H. Balakrishnan, 2002. An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wireless Commun., 1: 660-670.

Helal, S., W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, 2005. The gator tech smart house: A programmable pervasive space. Comput., 38: 50-60.

Idris, M.Y.I., E.M. Tamil, N.M. Noor, Z. Razak and K.W. Fong, 2009a. Parking guidance system utilizing wireless sensor network and ultrasonic sensor. Inform. Technol. J., 8: 138-146.

Idris, M.Y.I., E.M. Tamil, Z. Razak, N.M. Noor and L.W. Kin, 2009b. Smart parking system using image processing techniques in wireless sensor network environment. Inform. Technol. J., 8: 114-127.

Intanagonwiwat, C., R. Govindan and D. Estrin, J. Heidemann and F. Silva, 2003. Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Network, 11: 2-16.

Kai-Wei Fan, K.W., S. Liu and P. Sinha, 2007. Structure-free data aggregation in sensor networks. Mobile Comput. IEEE Trans., 6: 929-942.

Lindsey, S., C.S. Raghavendra and K.M. Sivalingam, 2002. Data gathering algorithms in sensor networks using energy metrics. IEEE Trans. Parallel Distrib. Syst., 13: 924-935.

Liu, Z., B. Wang and L. Guo, 2010a. A survey on connected dominating set construction algorithm for wireless sensor networks. Inform. Technol. J., 9: 1081-1092.

Liu, Z., B. Wang and Q. Tang, 2010b. Approximation two independent sets based connected dominating set construction algorithm for wireless sensor networks. Inform. Technol. J., 9: 864-876.

Poon, C.C.Y., Z. Yuan-Ting and B. Shu-Di, 2006. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. Communi. Magazine, IEEE, 44: 73-81.

Ratnasamy, S., B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin and F. Yu, 2003. Data-centric storage in sensornets with GHT, a geographic hash table. Mobile Networks Applicat., 8: 427-442.

Ruan, Z., X. Sun, W. Liang, D. Sun and Z. Xia, 2010. CADS: Co-operative anti-fraud data storage scheme for unattended wireless sensor networks. Inform. Technol. J., 9: 1361-1368.

Tovar, A., T. Friesen, K. Ferens and B. McLeod, 2010. A DTN wireless sensor network for wildlife habitat monitoring. Proceeding of the Electrical and Computer Engineering, May 2-5, Calgary, AB., pp: 1-5.

Wang, Y., T.L.X. Yang and D. Zhang, 2009. An energy efficient and balance hierarchical unequal clustering algorithm for large scale sensor networks. Inform. Technol. J., 8: 28-38.

Wang, W., B. Wang, Z. Liu, L. Guo and W. Xiong, 2011. A cluster-based and tree-based power efficient data collection and aggregation protocol for wireless sensor networks. Inform. Technol. J., 10: 557-564.

Yiwei, W. and L. Yingshu, 2009. Distributed indexing and data dissemination in large scale wireless sensor networks. Proceedings of 18th Internatonal Conference on Computer Communications and Networks, Aug. 3-6, IEEE Computer Society, Washington, DC., USA., pp: 1-6.

Zhang, W. S., G. H. Cao and T. La Porta, 2007. Data dissemination with ring-based index for wireless sensor networks. IEEE Trans. Mobile Comput., 6: 832-847.