

Data Management using Ring-based Index for Wireless Sensor Networks

Dasprakash.G
M.Tech (Student)
Computer Network and
Engineering
Acharya Institute of
Technology, Bangalore

Manjunath T N
Associate Professor
Dept of Information
Science and
Engineering Acharya
Institute of Technology,
Bangalore

Kanche Rajkumar
M.Tech (Student)
Computer Network and
Engineering
Acharya Institute of
Technology, Bangalore

ABSTRACT

In wireless sensor network (WSNs) network management is the process of managing, monitoring, and controlling the behavior of a network. The limitation of wireless sensor networks is storage power and data processing. Communication is the major source of energy consumption in WSN; various schemes have been proposed to reduce the internode transmissions. Strategies like data compression, in-network data processing better topology management, caching and indexing reduce the length of path traversed by data and or queries in the network. In this paper, we propose an index-based data scheme to address the problem

Keywords

Wireless sensor networks, System Model, Index-Based Data Dissemination, Adaptive Ring-based Index (ARI) Scheme

1. INTRODUCTION

A wireless sensor network (WSNs) is a device which monitors the physical and environmental conditions such as temperature, pressure, motion population etc. At different areas such sensor networks are expected to be widely deployed in vast area of environment for commercial, civil, military applications, habitat monitoring, intelligence and medical. These nodes are capable of measuring physical world, but also assessing, transferring, processing and storing, these measurements in the sensor itself. These nodes are battery powered and it's a wireless sensor nodes. Since bandwidth and power are scarce resources in wireless sensor networks. It is necessary to design scalable and energy efficient data schemes.

A several scheme was proposed like local storage, external storage and data centric storage-based in (DCS) where event can be detected by name and sensed data of these event are stored at the node with in the network instead of storing in external storage[1,2,3,4]. This technique used less power but unnecessary transfer of sensed data can be avoided in this external storage scheme using direct diffusion and the two-tier data dissemination scheme, in which source responds the data to the sink when sink has requested by query.

In TTDD[5], the source detecting the certain event and broadcast the event in the whole network, and sink interested in the event can send their queries directly to the source.

The proposed index based caching scheme. In this scheme, as shown in Figure1

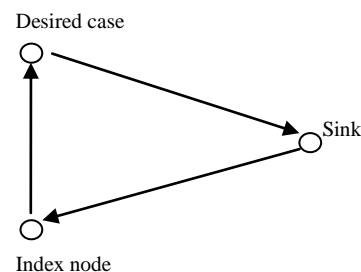


Fig. 1 Index based cooperative caching scheme

The sensed data of an event from the know target are stored within the nodes or neighboring nodes (called storing nodes). A storing node only sends data to a sink by a defined efficient path. When it receives a query request from the sink, the location information (called index) of the storing nodes are given to and maintained at some other nodes (called index nodes) based on the event related to the stored data. Hence, requested queries for a event can be routed to the appropriate index nodes where data is available. The index-based approach is more advantages than the existing data dissemination approach since it avoids both unnecessarily transferring the data and flooding the messages to the whole network. This approach has a disadvantage that it has to maintain an additional overhead for maintaining index nodes and which causes more power consumption and more bandwidth consumption. One major challenge in index-based data dissemination is how to maintain these indices in the network, such that the indices are made easily available to sinks whenever required and the index nodes are not overloaded. To overcome this problem, here an implementation based on the ring structure, and it can be called as the Adaptive Ring-based Index (ARI) scheme. In ARI, the index nodes for an event type are a set of nodes surrounding one or more particular region (called index centers of the event type). The index centers are found by applying hash function, e.g., Geographic Hash Table (GHT) [6], on requested event type. Note that the hash function maps an event type to one or more locations within the detecting region of the sensor network. The index nodes for the same event type are connected via some forwarding nodes to form a ring (called index ring). The number and locations of the

index nodes on an index ring, as well as the shape of the ring, can be adaptively changed to achieve load balance and optimize the system performance.

2. SYSTEM MODEL

Consider an application scenario:

A sensor network is deployed for monitoring many targets moving within a region. The sensor nodes in the network detect the status of each target, and periodically generate sensing data and it will store with in it or may be stored in separate node. Many users will be also moving within the region. From time to time, a user may issue a query via a sensor node (sink) for the data about the current status of a target and/or a summary of the recent status of the target. Assuming that the sensor nodes are stationary, and they will be knowing their own locations using GPS [7] or other techniques such as triangulation method [5]. To save power consumption, the nodes stay in the sleep mode most of the time based on the GAF protocol. Using this protocol, as shown in Figure 1, where each pair of nodes in neighboring grids can communicate directly with each other. Grid heads are responsible for forwarding the messages, and other nodes only need to wake up periodically.

Proposed data dissemination scheme is built a top of GPSR [9], a well-known geographic routing system for multi-hop wireless networks. GPSR uses two distinct algorithms for routing. One is a greedy forwarding algorithm which forwards packets progressively closer to the destination at each hop. The other is a perimeter forwarding algorithm that forwards packets where greedy forwarding is impossible. In this system, which uses both the GAF and the GPSR protocols, packets are forwarded by grid heads. A grid head first tries to forward a packet to its neighboring grid head that is closest to the destination. On failing to find such a node, it forwards the packet to one of the neighboring grid heads based on the perimeter forwarding algorithm.

In proposed data dissemination scheme, the targets to detect are classified into several types, each of which has a unique key. With certain hash function $H(\cdot)$, A key is mapped to one or more locations within the detecting region of the sensor network. a data centric storage scheme in this approach, relevant data is stored, by name, at nodes within the sensor net. Queries for data with a particular name can then be sent directly to the node storing that named data, thereby avoiding the flooding required in some data-centric routing proposals. The mechanism of data centric storage is that Data-centric storage is basically a distributed hash-table, supporting two basic primitives:

1. Put (key, value) this stores value (the data) according to the key (Which is based on the name of the data)
2. Get (key) this retrieves whatever value is stored at the key, (Which is derived from the event name?)

Given an event name, we hash that name into a key. This key is allocation somewhere in the sensor net. Thus, the hash function must be chosen with knowledge of the boundaries of the sensor net. The put (key, value) command sends a packet with the given pay load into the sensor net routed towards the location key. Our (slightly) modified form of GPSR will forward the packet to the sensor net node closest to this location, where the data can be stored; we will call this closest node the home node for an event. Similarly a get (key) command is forwarded to the sensor net closest to the key location, and that node then returns a packet to the source of the query with the corresponding data.

3. THE INDEX-BASED DATA DISSEMINATION

In this section, we describe the basic idea of the index-based data dissemination.

3.1 Basic Idea of the Index-Based Data Dissemination

The basic idea of the index-based data dissemination is illustrated in Figure 2

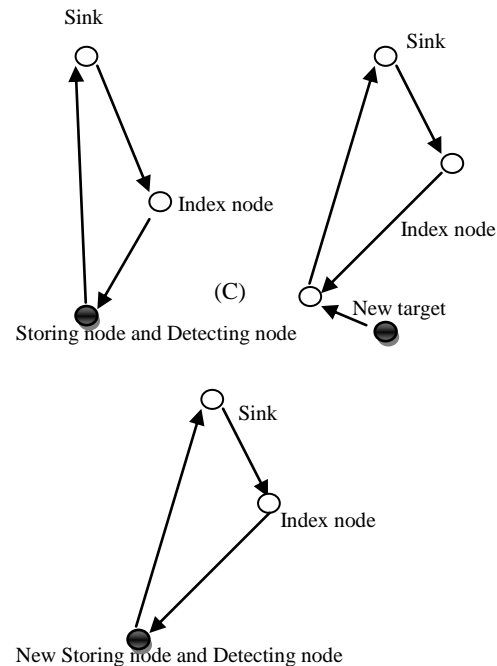


Fig. 2 Index-based data dissemination

A node detects a target periodically generates sensing data from that target, and stores the data in a storing node, within it or in neighbor node. When the target moves, as shown in Figure 2 (B) and (C), the detecting node will change accordingly. However, the new detecting node still stores the sensing data at the same storing node, until the storing node is far away from the detecting node, and then a new storing node is selected. When a new storing node of a target is selected, the old storing node processes the data it previously had stored and generates information of small size; sends the information to the new storing node. Note that the aggregation is not required by the index-based data dissemination scheme. If the previous data are not known to the new storing node, the new storing node may keep a pointer which points to the old storing node, such that the previous data can also be accessed from the current storing node. Also, the new storing node should register its location at the index nodes for the target. When a sink wants to query the sensed data of a target, it sends a request query message to an index node for the desired node in the region. On receiving the message, the index node forwards the request to the storing node which sends a response directly to the querying sink

4. AN ADAPTIVE RING-BASED INDEX (ARI) SCHEME

We now propose an adaptive ring-based index (ARI) scheme. We first present the motivations of the scheme, and then describe the operations of index querying/updating, failure management, and adaptive ring reconfiguration in detail

4.1 Motivation

The index-based data dissemination is similar to the problem of locating contents in the peer-to-peer systems such as Chord [10], Pastry [11], Tapestry [12], CAN [13], *etc.* In these systems, each node is assigned a numerical or a d-dimensional identifier, and it is responsible for owning pointers to objects (e.g., files), whose identifiers map to the node's identifier or region. The set of object in the nodes is altered as some nodes join or leave the system. Compared with these systems, sensor networks have their unique properties. For example, sensor nodes do not have unique id, but they can be aware of their own locations using GPS. Also, nodes may fail and are constrained in energy supply. Consequently, one challenge of implementing the index-based data dissemination is how to distribute, update, maintain and query the indices in the network, such that the following requirements are satisfied:

- **Fault tolerance:** Since sensor may fail, we should not depend on a single node to maintain an index. Techniques should be employed to achieve a certain level of fault tolerance by replacing the failed node.
- **Load balance:** Some index nodes may become overloaded. The indexing scheme should remove the overloaded index nodes and add some lightly-loaded index nodes in the grid, so the load is balanced with the other nodes.
- **Efficiency:** The scheme should not use too much overhead since bandwidth and energy are scarce resources in a sensor network.

One simple scheme can be derived directly from the basic DCS scheme. In this scheme, the index nodes for a target are the nodes which form the smaller region surrounding the location calculated. However, the index nodes cannot be moved to other node until the node is failed, and all queries and index updates for the target are processed by these nodes. Hence, the index nodes may become overloaded soon and reduces the efficiently, especially when the query rate or the index update rate for the target is very high this happens when a node fails. Also, since the index nodes for an event type are close to each other, the scheme cannot tolerate failures, which may destroy all the index nodes for an event type. A scheme similar to the structured replication DCS (SR-DCS) [6] or the resilient DCS (R-DCS) can be used to tolerate clustering failures. In these schemes, the whole detected region is divided into several sub regions, and there is one index node in each sub region. Using these schemes, the index nodes in some sub region can still be overloaded if the number of queries issued from that sub region is very high. In addition, a query may be routed through several levels of index nodes before reaching the valid index node. Tree-based approach, which is already applied in peer-to-peer networks, may also be used in sensor networks. However, this scheme may introduce significant maintenance overhead. For example, the failure of the root or some neighboring nodes in the tree may cause the tree to be reconfigured which causes the change in the tree. Furthermore, the nodes should be aware of the addition, removal and migration of index nodes. Thus, it is a challenge to design index-based schemes to satisfy the fault tolerance, load balance and efficiency requirements.

4.2 The ARI Scheme

The ARI scheme is based on the structure of ring, which includes the index nodes for the targets of the same type, and the forwarding nodes that connect the index nodes. The basic modules in the ARI scheme are ring initialization, index querying, index updating, failure management and ring reconfiguration,

4.2.1 Initializing an Index Ring

To initialize an index ring for the targets of a certain type. The index center for the target is calculated using a hash function [6], which finds the target to a location within the detecting region. Initially, the index nodes for the targets are m selected nodes whose distance to the index center is equal to or larger than r , where m and r are system parameters. These nodes are connected via some nodes to form an index ring surrounding the index center. Each node has two pointers which will be point to its clockwise and counterclockwise neighbor on the ring.

The index ring must satisfy the following conditions:

- (C1) The distance between the index center and any node on the ring with in the region should be larger than or equal to r .
- (C2) Any message sent by a node outside the ring with in the circled region and destined to the index center, will be intercepted by some node on the ring.

4.2.2 Querying and Updating an Index

The sink (S_i) can query the index of a target as follows:

It first calculates the distance to the index center of the target (L_k). If the distance is larger than r , the query message is forwarded along the direction of $S_i \rightarrow L_k$. Otherwise, the message is forwarded along the direction of $L_k \rightarrow S_i$. When a node receives a message, it performs differently based on the following cases:

- (1) If the node is an index node for the requested queried target: the query is forwarded to the storing node of the target if there exists such a storing node in the region.
- (2) If the node is a forwarding node on the index ring for the queried target: the query is forwarded to its clockwise neighboring node on the ring with in the region.
- (3) If the forwarding direction of the message is $S_i \rightarrow L_k$, and the distance between the node and the index center of the target is smaller than r : the forwarding direction of the query message is changed to be $L_k \rightarrow S_i$, and the message is forwarded in the new direction with new path.
- (4) Otherwise: the message is forwarded in the specified direction.

When the storing node of a target changes, the location of the new storing node must be updated at the index nodes for that target to get the data easily. The process of transferring an index update message to the index ring is similar to that of transferring an index query message. When the message arrives at an index node on the ring, the node updates its index, and forwards the message along the circle in the clockwise direction. If the node is a forwarding node, it is simply sent to the counterclockwise neighbor. The message is dropped when it is forwarded back to a node that has already received it and the process continues.

4.2.3 Dealing with Node Failures

To detect the failures of index nodes, neighboring nodes on an index ring should monitor each other by exchanging signaling or guiding messages periodically. In this paper, we consider the following failures:

Individual Node Failures

Due to insufficient energy or hardware failure, an individual node may fail in the network. This kind of failures can be expressed by GAF protocol, in which a failed grid head is detected and replaced by other nodes in the same grid. When a new grid head is selected after the old grid head (which is on an index ring) fails, the information (e.g., some indices and pointers) held by the old grid head is lost. However, the new head can receive signal messages from its neighboring nodes on the ring. From these signal messages, it knows that it is a node on the index ring, and can get the lost information from the neighbor's nodes. Due to some environmental condition, the nodes within a certain region may also fail or be isolated. This kind of failures is called clustering failures. Clustering failures may break an index ring and all stored information will be lost.

4.2.4 Ring Reconfiguration for Load Balance

Some nodes on the index ring become overloaded when compared to other neighboring nodes. For load balancing, these nodes should be replaced by some less-loaded nodes to perform the desired event task. In this scheme, by using algorithm to support load balance ring reconfiguration: Each node will be maintaining a flag to indicate whether it is like to be a node on an index ring. Initially, the flag will be made to turn on. When a node on the ring turns off its flag, it sends a quit message to its counterclockwise neighbor. On receiving that message, the neighbor initiates a process similar to the ring repairing process to remove the quitting node and reconfigure the ring with the new node.

5. CONCLUSION

This paper, proposes an index-based data dissemination scheme with adaptive ring-based index (ARI) which provides energy efficiency and less bandwidth is required. This scheme is based on sensed data that are collected, processed and stored at the nodes close to the detected nodes, and the location information of this node is pushed to some other index nodes within the target region. The index nodes for the targets of type form a ring surrounding the location which is determined based on the event, and the ring can be dynamically reconfigured to reduce the overhead. The ARI scheme should also be further enhanced to address other factors such as errors in localization and the existence of obstacles in the network.

As another important direction, we will study how to improve the efficiency of using storage space in the sensor network. This is essentially important when the network is expected to work for a long time and, thus, the generated data may use up the space affordable by the network. Some data compression and aging techniques should be developed to address the problem.

6. REFERENCES

[1] A. Ghose, J. Grobklags and J. Chuang. Resilient data-centric storage in wireless ad-hoc sensor networks. *Proceedings the 4th International Conference*

on Mobile Data Management (MDM'03), pages 45–62, 2003.

- [2] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy and S. Shenker. DIFS: A Distributed Index for Features in Sensor Networks. *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.
- [3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor network. *Proc. of the Hawaii International Conference on System Sciences*, January 2000.
- [4] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensor networks", Under submission to ACM SIGCOMM HotNets, Jul. 2002.
- [5] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks. *ACM International Conference on Mobile Computing and Networking (MOBICOM'02)*, pages 148–159, September 2002.
- [6] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage. *ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.
- [7] US Naval Observatory (USNO) GPS Operations. <http://tycho.usno.navy.mil/gps.html>, April 2001.
- [8] Y. Xu, J. Heidemann and D. Estrin. Geography Informed Energy Conservation for Ad Hoc Routing. *ACM MOBICOM'01*, July 2001.
- [9] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *The Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, Aug. 2000.
- [10] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. *ACM SIGCOMM*, August 2001.
- [11] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems. *The 18th IFIP/ACM Symposium on Operating Systems Principles*, August 2001.
- [12] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. *Technical Report UCB/CSD-01-1141*, 2001.
- [13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. *ACM SIGCOMM*, August 2001.