ISSN:0975-9646

Dasprakash.G et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5905-5911

An Study on Performance Analysis of Ring Based Index Data Management in WSN

Dasprakash.G[1], Manjunath T N[2], Kanche Rajkumar[3]

[1, 2, 3] *Dept of Information Science and Engineering,*
Acharya Institute of Technology, Bangalore

***ABSTRACT*--In Wireless sensor network management is the process of managing, monitoring, and controlling the behavior of each sensor network. The limitation of wireless sensor networks is limited storage space, limited battery power and limited data processing. Communication takes more source of energy consumption in WSN; various schemes have been proposed to reduce the internodes transmissions. Strategies like Data Compression, In-Network data processing better topology management; Caching and Indexing reduce the length of path travelled by data and or queries within the network. In this paper, we propose an index-based data scheme to solve the problem**

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

I. INTRODUCTION

A wireless sensor network (WSN) is a wireless network device using this device, we can sense and monitor physical or environmental conditions. It provides a bridge between the virtual physical and real worlds. Allow us to observe the condition at a fine resolution over large coverage scales .It has range of applications to security, industry, science, civil infrastructure and transportation. A WSN can be deployed in any location which consists of few hundreds to thousands of sensor nodes in a given location. The sensor node also includes a radio transceiver and receiver along with an antenna attached to it, a microcontroller, an interfacing electronic circuit board , and an energy source, usually a battery with small size capacity. The size of the sensor nodes may also vary from the size of a shoe box to as small as the size of a grain of rice. Hence, their prices also vary from a few rupees to hundreds of rupees depending on the sensor node used the functionality parameters of a sensor node are energy consumption, computational speed rate, bandwidth, and memory. These sensor nodes are capable of measuring real world, as well as assessing the information, transferring, processing and storing. The nodes are battery powered and it's a wireless sensor nodes. Since bandwidth and power are very limited in wireless sensor networks. It is necessary to design energy efficient data schemes and scalable. Several scheme are proposed like local storage scheme, external storage scheme and data centric storage-based in (DCS) scheme where event is detected by their name and the sensed data of these event will be stored in the node within the network instead of storing it in external storage [1,2,3,4].The technique uses more power and the query will be flooded to the neighboring nodes. Will lead to unnecessary transfer of sensed data from nodes can be avoided in this external storage scheme using direct diffusion scheme and the two-tier data dissemination scheme. In these schemes, a sink gets the data from the source when the sink has sent a query for the data. There need a sink-source matching mechanism to know that a sink has to find the source holding the data of the interest. The matching mechanisms is done by most LS schemes following a *flood-response* pattern [6], which has to be controlled flooding of messages. For example, in directed diffusion, a sink has to floods its requested query over the whole network; the source(s) with the requested data knows where to send the data. In TTDD Two-Tier Data Dissemination, the source detecting a certain event in the network floods the query of the event to the network, and the sink interested in the event can send their queries directly to the source. Considering the large number of nodes in the network, the network-wide flooding may introduce overloading of traffic in which source responds the data to the sink when sink has requested by query.

The proposed index based caching scheme. In this scheme, as shown in Fig:1

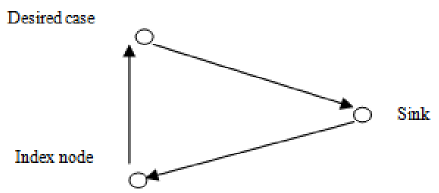

Fig: 1 Index based cooperative caching scheme

The sensed data of an event from the known target in the network are stored within the nodes itself or neighboring nodes (called storing nodes). A storing node May only sends data to a sink by a defined efficient path which was requested. When storing node receives a query request from the sink, the location information in the network (called index) of the storing nodes are given and maintained at some other nodes (called index nodes) in the network based on the event related to the stored data. Hence, requested queries for the event can be routed to the appropriate index nodes where data is available. The index-based approach has more advantages than the existing data dissemination approach because it avoids both unnecessarily transferring the data between the nodes and flooding the messages to the whole network. This approach has a disadvantage to that it has to maintain an additional overhead for maintaining index nodes in the network and which causes more power consumption and more bandwidth consumption which decreases the overall performance. One more major challenge in index-based data dissemination is to maintain these indices with in the network, such that the indices are made easily available to sinks whenever required and the index nodes are not overloaded at any time. To overcome this problem, an implementation based on the ring structure, is used called as the Adaptive Ring-based Index (ARI) scheme. In ARI, the index nodes for an event type consist of 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., Grid heads in the grid are responsible for forwarding the messages, and other nodes only need to wake up periodically for an event occurred or not. This scheme 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 the Greedy Forwarding Algorithm which forwards packets progressively closer to the destination at each hop nodes. The other is a Perimeter Forwarding Algorithm that forwards packets where greedy forwarding is impossible in the region. 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.

II. SYSTEM MODEL

Consider an application scenario, Sensor network is deployed in a location for monitoring moment of targets within the region. The sensor nodes in the network of that region detect the event of each target, and periodically generate sensed data and it will store with in it or may be stored in separate node in that region. Many users will be moving within the region in a network. From time to time, a user may issue a query at any moment via a sensor node (sink) for the data about the current event of a target and/or a summary of the recent event of the target. Each node will know their own locations using GPS [7]. To save power consumption, the nodes will be stay in the sleep mode most of the time based on the GAF protocol. Using this protocol, as shown in Fig: 2, where each pair of nodes in neighboring grids can communicate directly with each other as shown

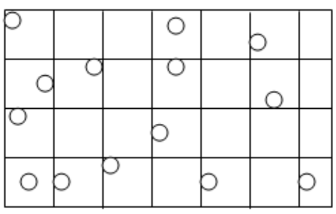

Fig: 2 Dividing a sensor network into grids

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 it has a distributed hash-table, supporting two basic primitives:

1. Put (key, value) is to stores value (the data) according to the key (Which is based on the name of the data)
2. Get (key) 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 given key. This key is allocated somewhere in the sensor network. Thus, the hash function must be chosen such that the boundaries of the sensor network. The put (key, value) command sends a packet with the given pay load into the sensor network routed towards the location key. Our (slightly) modified form of GPSR will forward the packet to the sensor net node closest to this location in the region, where the data can be stored itself; we will call this closest node the home node for an event for that region. Similarly a get (key) command is forwarded to the sensor network closest to the key location of the region, and that node then returns a packet that was requested to the source of the query with the corresponding data.

III. THE INDEX-BASED DATA DISSEMINATION SCHEME

The basic idea of the index-based data dissemination is illustrated in Fig: 3

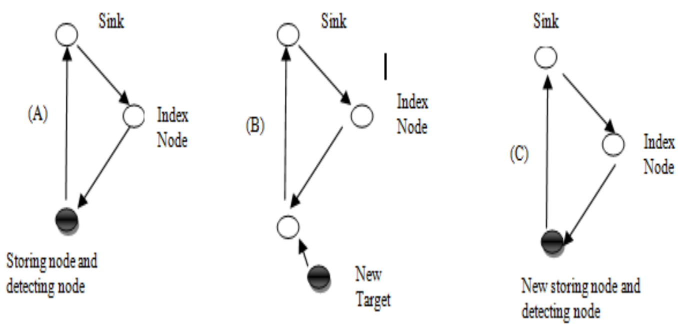

Fig: 3 Index-based data dissemination Scheme

A node may detects a target periodically generates sensed data from that target, and stores the data in a storing node, within it or in neighbor node. When the target moves within the target location, as shown in Fig: 3 (B) and (C), the detecting node will change accordingly. However, the new detecting node still stores the sensed data at the same storing node itself, until the storing node is far away from the detecting node, if it is far then a new storing node is selected. When a new storing node of a target is selected, the old storing node data is processed it previously had stored and generates information of small size to reduce the packet size; sends the information to the new storing node and newly sensed data. Note that the aggregation is not required by the index-based data dissemination scheme. If the previous data is not known to the new storing node, then 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 without losing its data. Also, the new storing node should register its location at the index nodes for the target to know In what region or location it is present. 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

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

We now propose an Adaptive Ring-Based Index (ARI) scheme. We present the motivations of this scheme, and then describe the operations of index querying/updating, failure management, and adaptive ring reconfiguration in detail. The index-based data dissemination as similar problem that 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 will be assigned a numerical or a d-dimensional identifier, and it has its own responsible for owning pointers to objects (e.g., files), whose identifiers map to the node's identifier or region. The set of object in these nodes will be altered as some nodes may join or may leave the system. When compared with these systems, sensor networks have their unique properties. For example, sensor nodes do not have unique id, but they can know 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 also should be satisfied:

- Fault tolerance: Sensor may fail on many issues, it 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 time 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 limited resources in a sensor network.

A scheme can be derived directly from the basic DCS scheme. In this scheme, the index nodes for a target in the region are the nodes which form the smaller region surrounding the location calculated. However, the index nodes cannot be moved to the other node until the node is failed, and all queries and index updates for the target are processed by these nodes only. Hence, the index nodes may become overloaded soon and reduces efficiently, especially when the query rate or the index update rate for the target is very high this happens when a node fails in the region. 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 will be one index node in each sub region. Using these schemes, the index nodes in some of the sub region can still be overloaded if the number of queries issued from that sub region is very high which causes more than limited. 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. Which makes the entire tree to reconfigure and more energy is required. 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.

I. The ARI Scheme

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

II. Initializing an Index Ring

To initialize an index ring for the target. The index center of the target should be calculated using a hash function [6], which finds the target to a location within the detecting region. Initially, the index nodes of the targets are selected nodes (m) whose distance to the index center are larger than or equal to r, where m and r are system parameters. These nodes will be connected via some nodes to form an index ring surrounding the index center. Each node will be having 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 equal to or larger than r.

(*C2*) Any message sent by a node outside the index ring with in the circled region and destined to the index center, will be intercepted by some node on the index ring

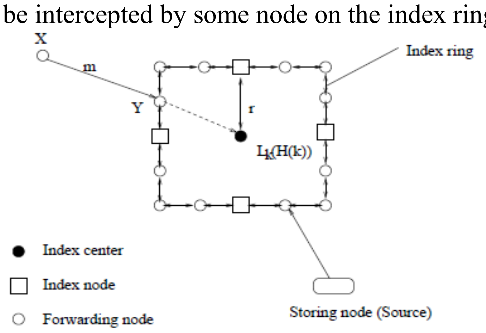

Fig: 4 Initializing an Index Ring

Fig: 4 shows an example of an initial index ring, where m = 4. The ring satisfies (*C1*), the distance between each node on the ring and the index center in the grid is at least r. Here each node can only forward messages to the other nodes in its neighboring grids a message sent by a node outside the ring- encircled region and destined to the index center, must pass some nodes in the ring on the ring. For example, as shown in Figure 3, message m which was sent by node X can be intercepted by node Y in grid. Thus, the ring also satisfies (*C2*). Note that, the index ring may be altered later, due to failures or load balance, but the resulted ring must also satisfy the above two conditions specified. m and r are the important Parameters affect the system performance. As m is large, the number of initial index nodes increases. Hence, the overhead for querying is decreased, at the cost that the overhead for storing and updating index is increased. When r becomes large, the overhead for queries issued by nodes far away from the index center is reduced. Also, more nodes are included in the ring, which improves the fault tolerance level. However, the overhead for queries issued by nodes within the grid region the ring is increased, as is the overhead for index update.

III. *Querying and Updating an Index*

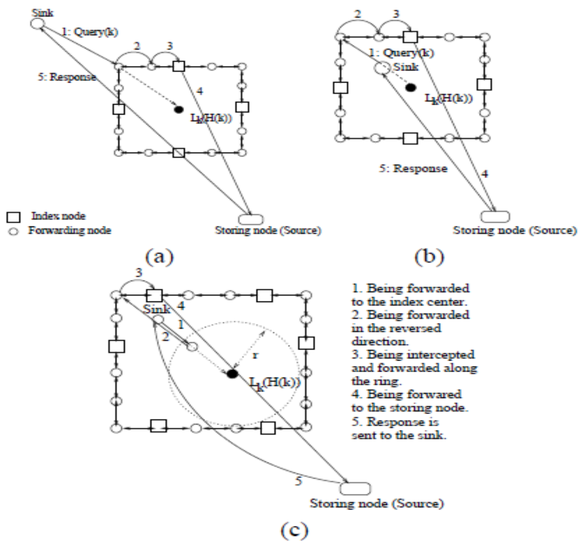

Fig: 5 showing querying and updating. The location of the new storing node must be updated at the index nodes for that target to get the data easily next time when needed. The process of sending an index updated message to the index ring is similar to that of sending an index query message. When the message arrives at an index node on the ring, the node updates its index information, and forwards the message along the circle in the clockwise direction. If the node is the forwarding node, it is simply sent to the counterclockwise neighbor. The message will be dropped when it is forwarded back to a node that has already received message and the process continues. The sink (Si) can query the index of a target as follows:

It first calculates the distance to the index center of the target (Lk). If the distance is larger than r, the query message will be forwarded along the direction of Si → Lk. Otherwise, the message will be forwarded along the direction of Lk → Si. When a node receives a message, it performs differently based on the following cases:

(1) The requested queried target, if the node is an index node for: The query will be forwarded to the storing node of the target if there exists such a storing node within the region.

(2) The index ring for the queried target**,** if the node is a forwarding node on: The query is forwarded to its clockwise neighboring node on the ring within the region.

(3) 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 Lk → Si, if the forwarding direction of the message is Si → Lk, and the message is forwarded in the new direction with new path.

(4) Otherwise: the message is forwarded in the specified direction only. Here the storing node of a target will be changing;

IV. *Dealing with Node Failures in the Grid*

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

Individual Node Failures in the Grid

Due to hardware failure or insufficient energy, an individual node may fail in the grid of the network. This kind of failures can be found by GAF protocol; first a failed grid head is detected and replaced by other nodes in the same grid to form a ring pattern. When a new grid head is selected after the old grid head (which is on an index ring) fails, the information which was (e.g., some indices and pointers) held by the old grid head will be lost. However, the new head can receive signal messages from its neighboring nodes on the ring to form a connection. 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 set of nodes within a certain region may also fail. This kind of failures is called clustering failures. Clustering failures will break the index ring and all stored information will be lost.

Clustering Failures:

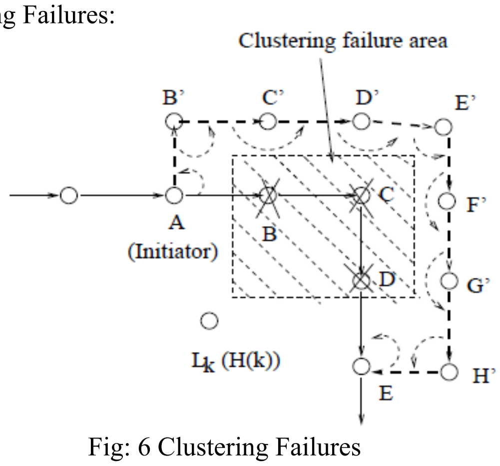

Fig: 6 Clustering Failures

Ring is broken after a set of node failure occurs. Here nodes B, C and D are failed and cannot be replaced by new nodes in their grids, since all the nodes in these grids are dead. The failure can be detected by the neighbors by beacon message of the failed nodes (i.e., nodes A and E), who do not receive beacon messages from the failed nodes for certain time interval. The counterclockwise neighbor (i.e., node A) of the failed nodes initiates a process to repair the ring. In the repairing process, at first A sends out a recovery message to the neighbors, and the message is forwarded around the failed region based on the *right hand rule* [9], which node is working, until the message arrives at a functioning node on the ring. The process is depicted in Fig: 6, After finding the new path the initiator A forwards the recovery message to B′, which is the next neighbor of A node and is sequentially counterclockwise from A from edge hA,Bi. On receiving the message to B', B' forwards the message to its next function neighbor C', which is the next one sequentially counterclockwise about B' from edge hB′,A′i. The process continues, until the message arrives at E, which is a functioning node on the index ring. When E receives the recovery message, E node will send an ack message back to A along the reversed path hE,H′,G′, F′,E′,D′,C′,B′,Ai. When A receives the ack message the new path is established, and the failed node of the ring, i.e., hA,B,C,D,Ei, is replaced by a nodes hA,B′,C′,D′,E′, F′,G′,Ei.

V. Ring Reconfiguration for Load Balance

Most of the time nodes on the index ring may become overloaded when compared to others nodes in the ring. For the purpose of load balance, these overloaded nodes should be replaced by some new lightly-loaded nodes in the ring. In the ARI scheme, we propose a simple algorithm to support load balance-oriented ring reconfiguration: Each node will be maintain a *willing flag* to indicate whether it is willing to be a node on an index ring. Initially, the flag will be turned on. When a node on the ring turns off says that the node is overloaded, now willing flag sends a quit message to its counterclockwise neighbor. On receiving the message, the neighbor initiates a process similar to the ring repairing process (refer to 4.2.3) it removes the quitting node and reconfigure the ring.

Table: 3 Lists most of the simulation parameters.

Channel type	Channel/Wireless Channel
Radio-propagation model	Propagation/Two Ray Ground
Network interface type	Phy/Wireless Phy
MAC type	Mac/802_11
Interface queue type	Queue/Drop Tail/ Pri Queue
Link layer type	LL
Antenna model	Antenna/Omni Antenna
Max packet in in freq	50
Number of mobile nodes	83
Routing protocol	GPSR
X dimension of topography	850
Y dimension of topography	850
Time of simulation end	30

V. RESULTS

We develop a simulator using ns2 (version allinone-2.32). In this NS2 simulator, the MAC protocol is used on IEEE 802.11, and the transmission range of each node will be 40m. Sensor nodes are distributed over 850*850 flat field, which is divided into 17*17m2 GAF grids, such that there will be one sensor node in each grid. For simplicity, we do not simulate the GAF protocol in the simulations.

Several targets are deployed in the detecting region. Each target in each grid may be moving in any direction. When a target enters a grid, the sensor node located in that grid can detect it. If the index-based data dissemination scheme is simulated, the sensor node that first detects a target becomes the initial storing node of the target. When the distance between the current detecting node of a target and the storing node of the target is higher than a threshold value, the storing node of the target is changed to be the current detecting node to reduce energy consumption. Any sensor node can be a sink in the grid which issues a query for the data of a certain target. Table: 3 lists most of the simulation parameters.

I. Simulation Results

Node Initializing, the index center for the target in the region is calculated using a hash function which will find a target to a location within the detecting region. Here nodes are distributed in the region. These nodes forms as shape of the ring as shown in Fig: 4, can be adaptively achieved to achieve load balance and optimize the system performance over the network. The field covered by the node is divided into small virtual grids as shown in Fig: 7.

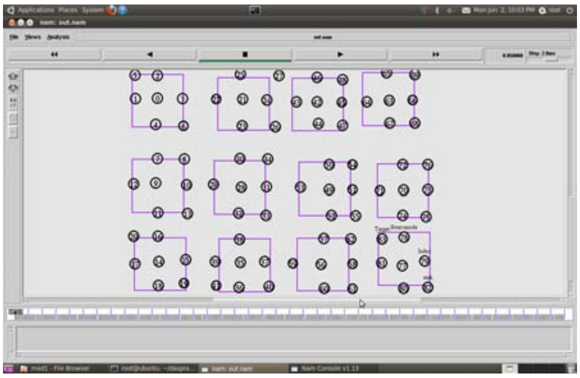

Fig: 7 Node is divided into small virtual grids

The virtual grid is defined such that, for any two adjacent grids, all nodes in one grid can communicate with all nodes in the other grid and vice versa. In each grid, each node will be rotating to keep awake and act as grid head in the grid, while others only need to wake up periodically to check there is any transition. The grid head is responsible for forwarding messages that pass through the grid. Data message or control messages are forwarded in the network, Greedy Perimeter Stateless Routing (GPSR) 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 our system, which uses both the GAF protocol and the GPSR protocol, 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.

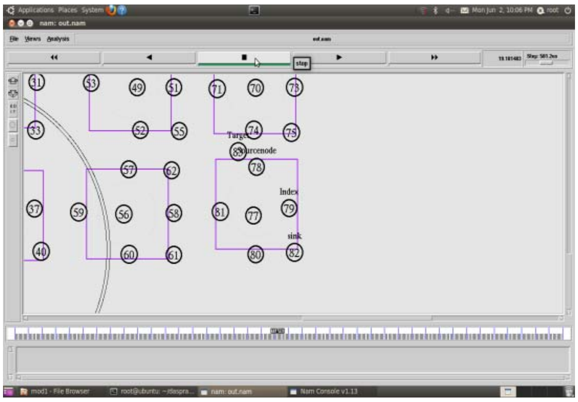

Fig: 8 The target changes continually

Showing the communication between the each nodes and the range of the each sensor nodes as shown in Fig: 8 as well as the location of the target changes continually.

Querying and Updating an Index

The requested queried target, if the node is an index node for: The query will be forwarded to the storing node of the target if there exists such a storing node within the region. Fig:9

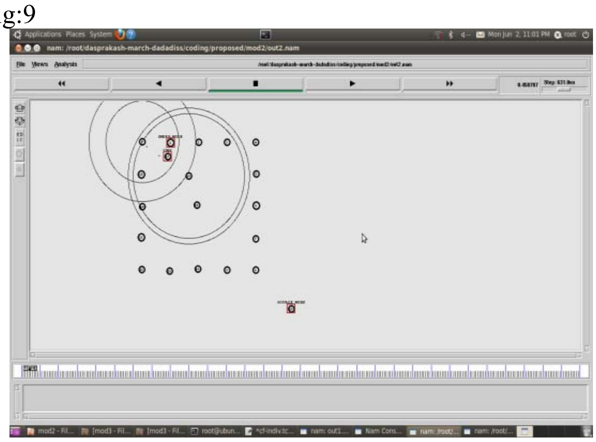

Fig: 9 Requested queried target

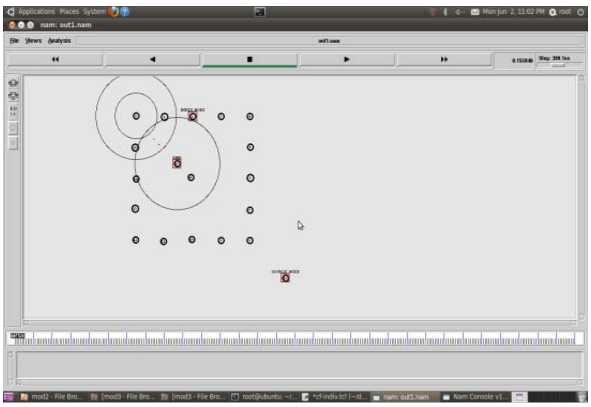

Fig: 10 Index ring for the queried target

(2) The index ring for the queried target**,** if the node is a forwarding node on: The query is forwarded to its clockwise neighboring node on the ring within the region. Fig: 10

(3) 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 Lk → Si, if the forwarding direction of the message is Si → Lk, and the message is forwarded in the new direction with new path. Fig: 11

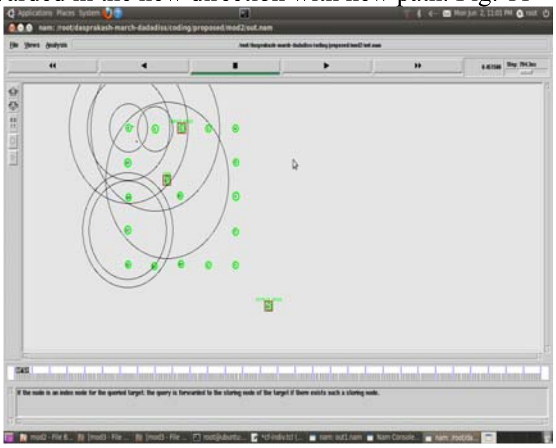

Fig: 11 Node and the index center of the target

Dealing with Node Failures in the Grid

When a new grid head is selected after the old grid head (which is on an index ring) fails Fig: 12.

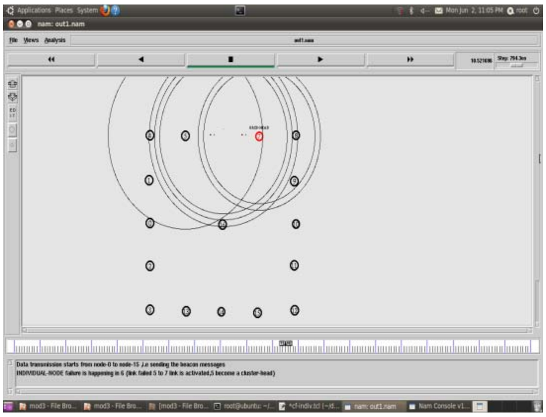

Figure 12 Individual Node Failures

Set of nodes failure and there new ring formation Fig: 13

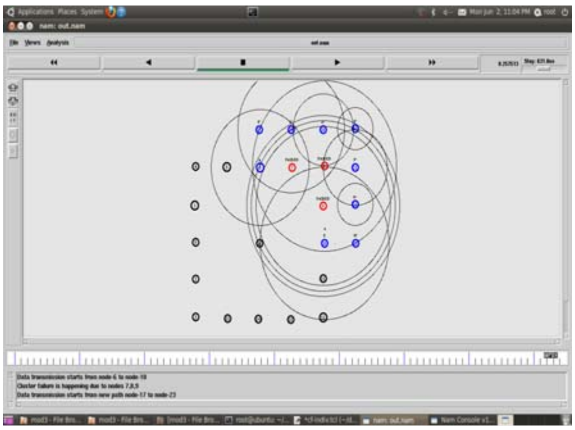

Figure 13 Clustering Failures

Comparison graph

Bit Error Rate

Fig: 14 shows that performance of the existing and proposed system .The graphs shows the Ring based Technique has less bit rate error then simple indexing

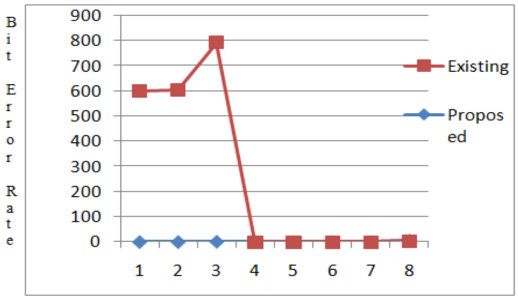

Fig: 14 Bit Error Rate

Control Over Head

Fig: 15 shows that performance of the existing and proposed system .The graphs shows the Ring based technique has less control over head then simple indexing which makes this more efficient

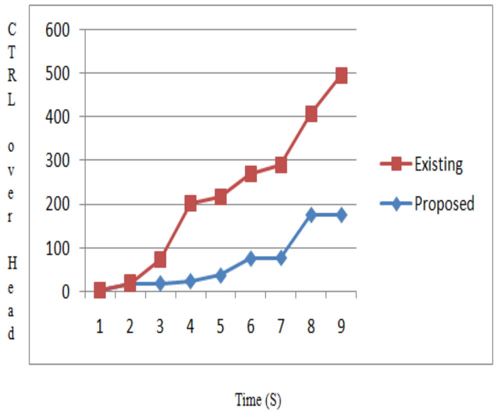

Fig: 15 Control Over Head

Throughputs

Fig: 16 shows that performance of the existing and proposed system .The graphs shows the Ring based technique has more packet delivery ratio then simple indexing hence less bandwidth usage

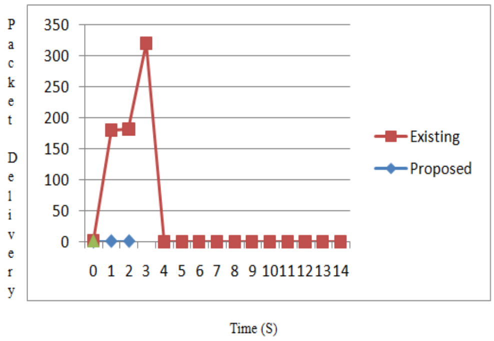

Fig: 17 Throughputs

VI. CONCLUSION

In this paper, we proposed an index-based data dissemination scheme with adaptive ring-based index (ARI). This scheme is based on the idea that sensing data are collected, processed and stored at the nodes close to the detecting nodes, and the location information of these storing nodes is pushed to some index nodes. The index nodes for the targets of one type form a ring surrounding the location which is determined based on the event type, and the ring can be dynamically reconfigured for load balance. Analysis and simulations were conducted to evaluate the performance of the proposed index-based scheme. The results show that the index-based scheme outperforms the ES scheme, the DCS scheme, and the LS scheme. The results also show that using the ARI scheme can tolerate clustering failures and achieve load balance.

REFERENCES

[1] Lu Li, Sai Ji, Tinghuai Ma and Jin Wang A Circle-Based Data Dissemination Algorithm for Wireless Sensor Networks with Mobile Sink International Journal of Smart Home Vol. 7, No. 1, January, 2013

[2] A. Ghose, J. Grobklags and J. Chuang.Resilient data-centric storage in wireless ad-hoc sensor networks.*Proceedings the4th International Conference on Mobile Data Management(MDM'03)*, pages 45–62, 2003.

[3] 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 SensorNetwork Protocols and Applications*, May 2003

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocal for Wireless Microsensor network. *Proc. of the Hawaii International Conferenceon System Sciences*, January 2000.

[5] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets", Under submission to ACM SIGCOMM HotNets, Jul. 2002.

[6] 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 Computingand Networking (MOBICOM'02), pages 148–159, September 2002

[7] 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 WirelessSensor Networks and Applications*, September 2002.

[8] US Naval Observatory (USNO) GPS Operations. *http://tycho.usno.navy.mil/gps.html*, April 2001.

[9] Y. Xu, J. Heidemann and D. Estrin. Geography Informed Energy Conservation for Ad Hoc Routing. *ACM MOBICOM'01*, July 2001.

[10] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *The Sixth AnnualACM/IEEE International Conference on Mobile Computingand Networking (Mobicom 2000)*, Aug. 2000.

[11] 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.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems. *The 18th IFIP/ACMSymposium on Operating SystemsPrinciples*, August 2001

[13] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An Infrastructure for Faul-tolerant Wide-area Location and Routing. *Technical Report UCB/CSD-01-1141*, 2001.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker.A Scalable Content-Addressable Network.*ACMSIGCOMM*, August 2001