Distributed Coordination of Sensors for End-to-End Reliable Event and Query Delivery

Nurcan Tezcan Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Email: {ntezcan,wwang}@eos.ncsu.edu

***Abstract*— In this paper, we present a new transport solution for WSNs addressing *bidirectional end-to-end event and query reliability*. We aim to reduce the reliable transport overhead while guaranteeing the reliability to deliver all events and queries in WSN applications. The proposed lightweight solution can achieve desired event reliability in conjunction with query reliability by operating with the least possible number of messages, and using a small subset of *coordinators* which are responsible of loss detection and recovery. Coordinators are selected using a *distributed, low-cost* algorithm with adaptive path discovery and maintenance features to utilize the cost of retransmissions and energy consumption. Simulation results show that, using such a coordination of sensors, significant savings on communication costs for event and query reliability are attainable while minimizing packet loss, energy consumption, and end-to-end delay.**

I. INTRODUCTION

Many applications developed for wireless sensor networks (WSNs) demand for reliable communication service, since the majority of these applications are event-critical applications. As sensor nodes gain more capability on reporting important perceptions in a collaborative fashion, more and more applications will be developed such as country border security, early fire detection, which are driven by the queries from the sink. Success of such mission critical applications are dependent upon the reliability of "information" delivery with underlying wireless networks. To understand the problem of reliability in this context, we need to elaborate on the following question: "What is the information to be delivered reliably in WSNs?"

In conventional reliability context, transport service has no additional knowledge on the semantics of the information, thus reliability solutions are per transport message segment based (shortly, message-level) [1], [10]. In such transport solutions, end-to-end reliability ensures that each message is individually received by the intended end point successfully. However in WSNs, information of interest is carried into an event which is usually transfered with more than one transport message segment due to the overlapping sensing ranges of many sensor nodes.

Due to above reasoning, a conventional message-level reliability would involve reliable delivery of many redundant event messages in a WSN. This is a very fundamental challenge not only from the perspective of energy conservation, but also from the perspective of delivery latency under congested network conditions. In message-level reliability, many redundant event reports have to be retransmitted even in case of congestion which can make the network more unstable, energy wasting, and potentially nonoperational. Hence, a reliable delivery mechanism must provide reliability by operating with the least possible number of transport segment messages in a WSN. In order to achieve such an objective both for queries and event report messages, we carefully distinguish *event reliability* and *query reliability* as follows.

Event reliability is defined to be achieved when every critical event report message is received by the sink node. This is the necessary and sufficient condition for sensor-to-sink direction reliability. *Query reliability* is defined to be achieved when every query of the sink is received by those sensors that cover the entire sensible terrain within the area of deployment, which is necessary and sufficient for sink-to-sensor direction reliability.

An effective technique to achieve such an event and query reliability would involve a *distributed coordination* of sensors to form a network topology that is sufficient to cover the entire sensible terrain. The distributed coordination is motivated by the idea of using randomly deployed sensor networks to form an impromptu network in an energy-efficient way without presuming a particular geographical distribution or location-awareness. If such a topology is constructed, sensor nodes will be mapped to a small set of *coordinators* that can collect data and forward data to the sink. In this case, the process of reliable event delivery will be beneficial, since only coordinators can participate in message-loss detection and recovery. Effectiveness of such an approach is two-fold. Reliable event delivery mechanism has to process a subset of the information carrying event report messages, allowing reduction in processing, spectrum, and energy resource consumption. Further, dealing with only a subset of the large number of sensor nodes reduces the algorithmic complexity of the reliable delivery mechanism when lost message recovery would be needed.

Much of the prior work on reliable transport in WSNs addresses message-level reliability on either upstream (sensor-to-sink) or downstream (sink-to-sensor) reliable data delivery. *Pump Slowly, Fetch Quickly* (PSFQ) [9] is proposed for downstream reliable data delivery from source to the sensor nodes. It is based on hop-by-hop error recovery with in-network caching and sending repair request via NACKs (Fetch) that is faster than the source transmission rate (Pump). In *GARUDA* [4], downstream message reliability is achieved by a virtual infrastructure that is composed of local and designated loss recovery servers. *Reliable Multi-Segment Transport* (RMST) [6] is a transport layer protocol which is designed to run in conjunction with directed diffusion. It is a selective NACK-based message-level reliability scheme used for transfer large amount of data from sensors to sink.

In terms of event reliability, *Event-to-Sink Reliable Transport* (ESRT) [5] is motivated by the fact that the sink is only interested in reliable detection of event characteristics from the collective information provided by sensor nodes. Although it considers the event information, it is designed for sensor-to-sink communication. However, considering only upstream or downstream reliability is not sufficient and it will restrict the potentials of a network protocol because reliable transmission service is fundamental to both sensor nodes and the sink. Hence, two-way reliability feature should be provided in a single transport mechanism. Among transport layer mechanisms proposed for sensor networks [4]–[6], [9], to the best of our knowledge, our attempt is the first that makes use of *event reliability* in conjunction with *query reliability*. Our simulation study helps us to validate our main goal by showing that the cost of reliable service using a coordinated sensor network is reasonably low in order to achieve event and query reliability.

The remainder of the paper is organized as follows. Section II gives some assumptions and challenges that will be used in the remaining of the paper. In Section III, we propose the distributed coordination algorithm. We present the design of reliability schemes for end-to-end event and query delivery in Section IV. Performance evaluation is discussed in Section V, and Section VI concludes the paper.

II. PRELIMINARIES

In this paper, the sensor network is modeled as a directed graph $G(V, E)$, where V is the set of vertices ($|V| = N$), representing the sensor nodes, and E is the set of edges, representing the communication links. All links are directed. We also consider the fact that links may be asymmetric due to channel conditions and radio irregularities [13]. A communication link is symmetric if there exists a link from v_i to v_j whenever a link from v_j to v_i exists. One way to determine whether the link is symmetric or not is to exercise the *neighbor discovery* scheme given in [13].

After the deployment, each node sends periodic beacon messages to find its one-hop neighbors. While sending beacon messages, sensors piggyback the IDs of all nodes they received beacons from. In this process, node v_i marks the link towards node v_j to be *symmetric* if its own ID is listed in the beacon message received from node v_j, implying (v_i, v_j) and (v_j, v_i) $\in E$. Sensors v_i and v_j are neighboring nodes only if there exists a *symmetric* link between them. The neighbor set of node v_i is represented by $\mathcal{N}(v_i)$.

We assume sensors to be static and *location-unaware* having fixed transmission power. Each sensor v_i has a unique identifier such as a MAC address and its initial battery energy is given by $e_i(0)$. We assume that sensors have the ability to monitor their residual energy. *Residual energy ratio* of node v_i at time t, denoted by $w(v_i, t)$, is calculated as:

$$w(v_i, t) = \frac{e_i(t)}{e_i(0)}, \tag{1}$$

where $e_i(t)$ is the residual energy of the sensor at time t.

We first choose a set of *coordinator nodes*, denoted by *CNs*, covering the entire field, where the rest of the *sensor nodes* are denoted by *SNs*. Each node is either a coordinator or can directly communicate with a coordinator (via one-hop). Coordinators will be responsible for the reliable delivery of events within their sensing ranges and their one-hop neighbors' sensing ranges.

We now summarize the following key challenges for a distributed topology that can provide reliable communication.

Energy-efficiency: Although the our primary goal is reliable event and query delivery, we aim to accomplish this with minimum possible energy expenditure. Hence, all nodes can not participate to event and query reliability. Therefore, we need to select a subset of coordinator nodes which helps to form a more scalable and energy-efficient network. A coordinator acts not only as a data aggregation point for collecting sensing data from sensors, but also as a loss/recovery server to make various retransmission decisions.

Network dynamics: Reliable event and query delivery must be established and maintained in despite of dynamic topology in WSNs. Topology dynamics can result from either the failure or temporary power-down of energy constrained sensor nodes.

Minimizing end-to-end delay and bandwidth usage: The proposed reliability solution should reduce the end-to-end delay while guaranteeing the reliability to deliver all events and queries that WSN applications require. Since sensor have limited resources, we need to accomplish the reliability by working with the least possible number of messages.

Node heterogeneity: Sensor nodes may not be identical in terms of their energy reserves or their transmission range due to local transmission conditions. Thus, the protocol should be adaptive to local conditions, thus neighborhood discovery or energy-related operations should be held in a distributed way while the network topology is constructed.

Scalability: The number of sensors deployed in a sensing field may reach large numbers. Thus, our protocol must utilize the high density nature of the sensor networks.

Next, we will explain the distributed coordination mechanism and its features in detail.

III. DISTRIBUTED COORDINATION OF SENSORS: OBJECTIVES AND ALGORITHMS

The overarching goal of our design is to maintain the network topology while achieving a certain degree of event and query reliability in the most efficient way. Advantages of this topology lie in reliability and energy efficiency dimensions such that: (i) only a set of sensors (coordinators) will be responsible in loss detection and recovery to provide event and query reliability; (ii) sensor nodes transmit information only to their coordinators (over a relatively short one-hop link), and those coordinators aggregate the received information to be forwarded towards the sink node.

Procedure to establish the distributed coordination involves three steps: (i) *constructing a coordinating set of sensors* with appropriate costs; (ii) *discovering query-specific paths* incurring low cost forwarding nodes; and (iii) *maintaining coordinators* to handle cost changes, node failures and joining. We first explain the distributed coordination algorithm in detail, and then summarize path discovery and maintenance features.

A. Distributed Coordination Algorithm

Here we present a distributed, low-cost algorithm for the coordination problem. We select a subset of sensors as coordinators/loss-recovery nodes during a coordination period using one-hop neighbor information. Coordinator selection is primarily based on the residual energy of each node and the degree of connectivity, which is called *cost* of the sensor. The cost of a sensor node v_i at time t, denoted by $c(v_i, t)$, is given as the total energy expenditure of node v_i normalized by the number of its neighbors:

$$c(v_i, t) = \frac{1 - w(v_i, t)}{|\mathcal{N}(v_i)|}, \qquad (2)$$

where $|\mathcal{N}(v_i)|$ is the number of neighbors assuming all nodes have at least one neighbor, and $w(v_i, t)$ is defined in (1). Due to our objective of selecting sensors of minimum cost, those sensors with high degree of connectivity and high residual energy reserves are more likely to be in coordinating set than others.

Our goal is to coordinate the network successfully within a predetermined period, called *coordination period*, T_p. We assume that sensors have globally synchronized clocks [2]. Coordination period start by a synchronization message flood started by the sink after deployment. Drifts in the reception times of this synch message is relatively small compared to T_p. During the coordination period, each sensor announces itself as a CN in its *flag-time*, denoted by T_{flag}. The flag-time of a node is proportional to its cost value:

$$T_{flag}(v_i, t) = c(v_i, t) \cdot T_p, \qquad (3)$$

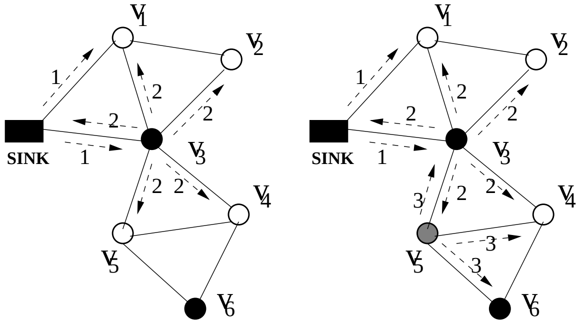
where $c(v_i, t) \in [0, 1)$ is the cost of the node given in (2) and T_p is the coordination period. The rationale behind (3) is that intuitively, sensor nodes that have lower cost are desired to be coordinators. In other words, these nodes should send coordinator announcement message earlier by having shorter waited flag-times.

After the initial deployment, sensors know their own costs, flag-times, given in (3) and neighboring sets. Initially, all sensors are neither SN nor CN. Distributed coordination algorithm is based on first-come first-serve approach given in **Algorithm 1**. With the start of coordination period, node v_i, having the minimum cost, thus shortest flag-time, sends *I-am-CN* message and becomes a coordinator node. When neighbors of v_i receive this *I-am-CN* message, they automatically become SNs, and thus immediately send an *I-am-SN* message to all their one-hop neighbors to announce its new role. Note that, *flag-time* is dynamically updated during T_p by using (3). Number of neighbors in the cost function indicates the neighbors which are neither a CN nor SN, thus after receiving a SN announcement message, sensors update their neighbors to have an accurate flag-time.

Algorithm 1 Distributed Coordination Algorithm

```
Compute and broadcast $w(v_i, t)$
$c(v_i, t) \leftarrow \frac{1-w(v_i,t)}{|\mathcal{N}(v_i)|}$
$T_{flag}(v_i) \leftarrow c(v_i, t) \cdot T_p$

Repeat
  if (I-am-CN is received from $v_j \in \mathcal{N}(v_i)$) {
      broadcast (I-am-SN); CN $\leftarrow v_j$; break;}
  if (I-am-SN is received from $v_k \in \mathcal{N}(v_i)$) {
      $|\mathcal{N}(v_i)| \leftarrow |\mathcal{N}(v_i)| - 1$; update $T_{flag}$;}
  if ( $(t == T_{flag}(v_i))$ && no I-am-CN is received) {
      broadcast (I-am-CN); break; }
Until $(T_p)$
```



(a) After receiving from the sink, sensor v_3 broadcasts the query to its neighbors.

(b) Next, sensor v_5 broadcasts the query and become an FN.

Fig. 1. Illustration of path discovery algorithm.

At the end of T_p in **Algorithm 1**, all nodes either send an *I-am-CN* message on its flag-time and becomes a CN, or receives an *I-am-CN* message before its flag-time and become an SN with an associated CN. Assume that node v_i has the maximum cost value $c(v_i, t) \in [0, 1)$, thus the maximum $flag\text{-}time \in [0, T_p)$. In the worst case, node v_i will wait during its flag-time without hearing from any CN and declare itself a CN before the coordination period ends.

B. Dynamic Path Discovery Algorithm

In conjunction with the coordination, we introduce a *query-specific* dynamic route discovery algorithm to find the *forwarding nodes*, denoted by *FN*, which will relay the messages through CNs, in case CNs are not transmission range of each other. In sharp contrast to earlier clustering studies [11], [12], query-specific forwarding nodes are selected dynamically during the query transmissions from the sink. Nodes use links which are found to be *symmetric* in neighbor discovery stage. This helps in reusing the same path for both downstream and upstream flows. Once queries are received by CNs, the event reports will be relayed over the same routes used for distributing those queries.

Step 1: Path discovery starts by the sink broadcasting a query message. First, one-hop neighbors of the sink that receive the query, have to decide on forwarding this query. Whenever a node forwards the query to its neighbors, this node becomes an FN immediately. Therefore, the idea is to wait for an overhearing period to observe that being an FN is necessary to achieve the connectivity among CNs or not. By overhearing its neighbors, node v_i confirms that the query has been received by all its neighbors, thus, another broadcast from v_i would be useless.

Step 2: All CNs are also forwarding nodes, thus when a CN receives a query, it will immediately broadcast the query. Otherwise, sensor node needs to wait for a short *overhearing-period*, denoted by T_{hear}, to decide on forwarding.

Step 3: At the end of T_{hear}, v_i will broadcast the message unless it overhears that this query has been received and forwarded by its own coordinator, or all of its SN neighbors. When an SN broadcasts a query, it becomes a FN for the next round of query transmissions. We determine and update T_{hear} dynamically based one one-hop packet transmission delay.

Above mechanism determines the forwarding nodes during the distribution of the query, instead of using a second coordination stage that may incur extra time and message complexity. Thus, paths between the sink and CNs are discovered dynamically which is adaptive to topological changes. Next, we will explain how a low-cost coordination topology maintenance is achieved.

C. Distributed Topology Maintenance Algorithm

After selecting an initial set of CNs effectively, a challenge is how to maintain this network topology to handle expected or unexpected node failures and provide fair energy expenditure among sensors. In general, topology maintenance can be classified as: *global* and *local* maintenance.

In *global maintenance* attempt, the coordination must be repeated periodically independent of the current topology to be able to permute CNs. In particular, global maintenance is the process of rerunning the distributed topology creation algorithm with the latest cost values of the sensors. This ensures a timely switching of CN and SN roles among the sensor nodes to maintain a uniform energy expenditure throughout the network. However, global maintenance can not handle the unexpected node failures between two coordination periods.

On the other hand, *local maintenance* can be triggered in case of unexpected CN failure. When SNs detect the failure, they select their new CN(s) to retire the failed. This is done by invoking the distributed coordination algorithm for these particular SNs. It may not be possible to find a single coordinator to replace the failed CN; nonetheless, the overall cost of self-healing would be less than the cost of triggering a global maintenance for the entire network.

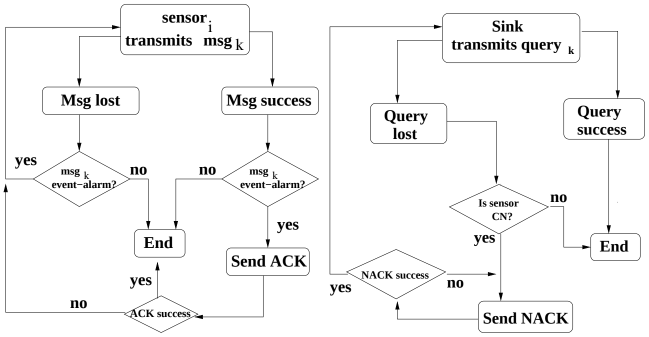
In this work, we combine reactive local maintenance with periodic global maintenance to bring down the frequency of global maintenance needs. Striking a good balance between these two approaches is the key to creating an effective low-cost topology maintenance. Next, we will explain how end-to-end reliability is achieved using coordinators.

IV. RELIABLE EVENT AND QUERY DELIVERY

Reliability of WSNs is categorized as *event and query* delivery reliability, whereas the least possible number of messages are transmitted in order to achieve energy conservation and low delivery latency. Therefore, we need to clearly define event and query reliability notions in WSNs for sensors-to-sink and sink-to-sensors data delivery.

End-to-end reliable event transfer is achieved when the first message indicating the event (sent by CNs) is successfully received by the sink. Note that sensors may send more than one message indicating the same event, even though the successful delivery of the first message is sufficient to achieve the reliable delivery of desired event. Similarly, *end-to-end reliable query transfer* is referred to as all queries are received by coordinator nodes successfully.

To achieve end-to-end query and event reliability, we propose a novel coordinator-based loss detection and recovery mechanism. Our goal is to avoid extra cost of reliable transport services, and to guarantee the minimal reliability to deliver all events and queries that WSN applications require. In this approach, reliable event and query delivery service is controlled by CNs where only enough retransmissions are done to satisfy the loss event or query.



(a) Sensors-to-sink. (b) Sink-to-sensors.

Fig. 2. Reliable event and query delivery.

A. Reliable Query Delivery

Reliable query delivery (sink-to-sensors) is provided using negative acknowledgements sent from coordinators to the sink if there is a query loss. Since the queries sent by the sink are in order, CN can detect the lost message by use of *sequence numbers* in the query messages. When a CN detects a gap in the sequence number of the new query, it sends an NACK back to the sink to recover the previous query. If a lost query is detected and the node is a coordinator, it sends an NACK until it receives the query. Otherwise, it simply ignores the sequence numbers.

In this scheme, lost query messages can be detected when CNs receive a new query message. This may result high latency if the query transmission frequency is low or the lost query is the final query sent by the sink. Consider the last message q_k with sequence number k is lost. CN may not handle the lost message since there is no consecutive query. To differentiate the final query message, we use an extra *Poll/Final* (P/F) bit which can be set by the sink node. P/F bit is set either when a message is the last query or the next query will be sent after timeout. Therefore, CNs which receive a query with P/F bit set send an ACK to the sink, indicating the query is received successfully.

B. Reliable Event Delivery

The NACK mechanism based on sequence numbers does not work efficiently for reliable event delivery because event information is sent by individual sensors and it is usually out of sequence. Hence, NACKs cannot handle the lost event messages by finding the gap in sequence numbers. However, using an ACK mechanism that requires acknowledgement for each message may result inefficient use of battery power, which is considered to be a very scarce resource in WSNs.

Hence, we propose a *lightly-loaded ACK* mechanism between the CNs and the sink node. Each CN waits for acknowledgement for only the first message that reports an event, i.e., *event-alarm*. When a new sensing value is obtained, a CN decides if it reports an event or not. If it is an *event-alarm*, it simply marks the message by setting the *Event Notification* (EN) bit. Therefore, the sink node sends ACK for the only messages which are marked as event-alarm. Similar to downstream communications, only the CNs are responsible for waiting the acknowledgement and may retransmit if necessary.

While using NACK, the sink only retransmits when it receives NACK for query delivery. Therefore, no timer is used. However, while transferring events from sensors to the sink, CNs wait ACKs for event-alarm messages. When an CN sends an event alarm message, it starts the timer and waits for *timeout period* (t_{out}) to retransmit, which is dynamically determined based on *round trip time* (RTT) similar to adaptive retransmission timeout in TCP.

We assume that all sensors have an initial RTT which is the duration between the time when a message is sent and the time when the ACK of the message is received at the sender. Then, we calculate $RTT(sample)$ dynamically which refers to the latest RTT. $RTT(sample)$ is calculated using the *time stamp* field. Sensors assert the time information in their messages sent back via ACK by the sink. Thus, CNs can determine the $RTT(sample)$ by comparing the time stamp received by ACK.

V. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the proposed scheme via simulations in ns-2 [3]. We conducted several simulations using different scenarios in a static sensor network. We first explain the simulation setup and parameters, then discuss the results.

A. Simulation Setup and Parameters

Simulations are performed for randomly placed sensor nodes in a rectangular region. All sensor nodes have a *sensing region* of fixed range, r, associated with them. A communication edge exists between two sensors nodes if they are within their transmission range. A sensing field of 250 x 250 m^2 is used in simulations. We vary the number of sensors which allows us to study the performance from very sparse to very dense networks. The number of sensors should be sufficient to cover the sensing field for given parameters. In the basic scenario, 100 fixed sensor nodes having transmission range of 90 m and initial energy of 5 J are used.

In the experiments, we use a mobile tracking application in which the movements of mobile nodes are reported to a sink. Mobility pattern of a mobile (phenomenon) node is generated using Gauss-Markov mobility model [8] at a maximum speed of 20 m/sec. An event is defined to detect the phenomenon node in the sensing area of a sensor.

We follow an *event-driven* data delivery model to transfer data from sensors to the sink. Sensors send data only if they detects an event. If an event is detected in the period of an update interval, a sensor reports the event to the sink by sending consecutive messages. We use the parameter *event-reporting frequency*, denoted by f_e, to customize how frequently a sensor node sends event reports when phenomenon is in its sensing area. Note that, the first report is regarded as the event alarm message. On the other side, the sink uses a *continuous* data delivery model, by sending periodic queries to the sensors. Similarly, we use *query-reporting frequency*, denoted by f_q, as a simulation parameter to maintain traffic load in downstream direction. The coordinates of the sink is the center of the sensing field and same for all experiments.

B. Simulation Results

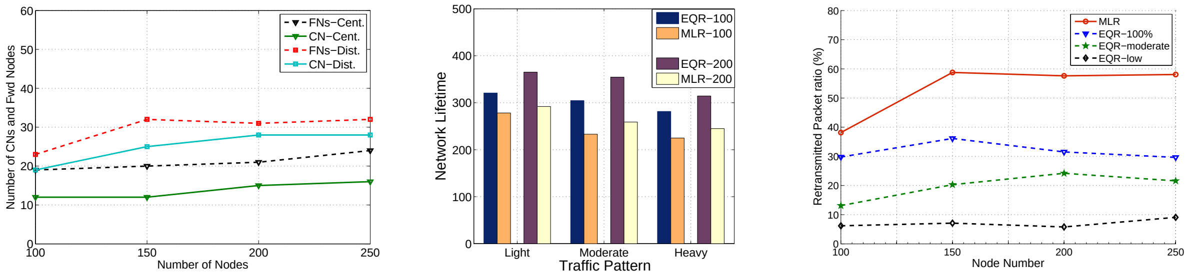
The performance of the proposed mechanism is evaluated regarding the effectiveness of coordination algorithm and its maintenance, dynamic path recovery and query and event reliability. In this context, we refer the proposed Event and Query Reliability mechanism as *EQR* and Message-Level Reliability as *MLR*. We start by illustrating the effectiveness of the coordination algorithm (**Algorithm 1**), i.e., the number of coordinators, energy efficiency and network lifetime.

We first measure the number of coordinators (CNs) and forwarding nodes (FNs) in distributed coordination for different network densities in Fig. 3 (a). Our goal is to discuss how proposed distributed coordination algorithm performs so that how many sensor nodes will work for loss and recovery of missing events/queries to reduce overhead. We then compare the results with a centralized algorithm [7] that uses the sink as an authority to select CN set with the same *cost* function assuming all locations and residual energy of sensors are known. It is intuitive that having the global view of the network at the sink node provisions algorithms for near-optimal solution determination. Based on this rationale, in Fig. 3 (a) shows that number of coordinators in distributed coordination is at most 10% compared to the centralized approach. In a low density network at most 20% of nodes work as a CN and responsible for loss detection and recovery which might reduce the retransmission overhead considerably. Furthermore, as the number of nodes increases, there is no significant change in the number of coordinators. Therefore, the proposed coordination algorithm can be efficiently used in reducing the cost of reliability in dense networks.

In Fig. 3 (b), we plot network lifetime against different traffic patterns for 100 and 200 number of nodes. We have simulated three types of traffic load scenarios: (i) heavy: $f_e = 0.1$ and $f_q = 2$ and (ii) moderate: $f_e = 0.5$ and $f_q = 5$ (iii) light: $f_e = 1$ and $f_q = 10$ sec. We consider a WSN as *alive* when the sensing field is fully covered. According to this, we observe that network lifetime is prolonged significantly using *event and query reliability* compared to *message level reliability*, especially in high density networks. Even in low density networks, network lifetime is prolonged up to 15% which shows the effective energy savings of event and reliability approach.

Further, we show the retransmission ratio in Fig. 3 (c). While achieving the end-to-end reliability from sensors-to-sink and sink-to-sensors, the ratio of retransmitted messages changes dramatically with the required reliability level. As shown in Fig. 3 (c), the retransmission overhead ratio of MLR is reduced up to 30% compared to the retransmission ratio of 100% EQR in distributed coordination. We also show different EQR levels as *EQR-moderate* and *EQR-low* by limiting the number of retransmissions such that, lost event or query is retransmitted once in EQR-low. Therefore, when the WSN application depend upon low reliability but stringent energy constraint, reliability can be tuned to degrade the retransmission overhead up to 25%.

Fig. 4 and 5 show the performance of our distributed coordination algorithm and path discovery scheme with respect to



(a) Average number of coordinators.

(b) Network lifetime of distributed coordination.

(c) Ratio of retransmitted packets to the total number of packets.

Fig. 3. Effect of distributed coordination.

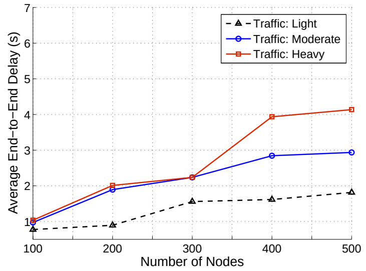


Fig. 4. Average end-to-end delay for various traffic patterns.

average end-to-end delay and packet loss ratio. From Fig. 4, we find that the end-to-end delay is a function of increasing network density. End-to-end delay in distributed coordination degrades gracefully with decrease in traffic load. Even in heavy packet load, the delay remains below 4 sec.

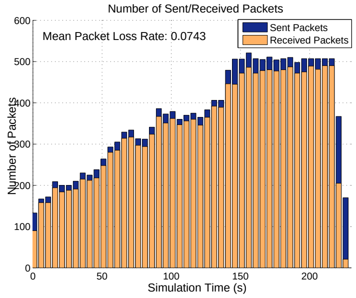


Fig. 5. Sent/received packets.

In Fig. 5, we illustrate the number of received packets overlapped to the number of sent packets, where the visible dark portion indicates the difference between sent and received packets, thus packet loss. In event-driven applications, the total number of packets sent to the sink are subject to event-driven traffic, thus changing over time. Hence, we depict the sent and received packets in time and observe that the average packets loss ratio is 7% when 100 nodes are deployed. Based upon the random deployment of nodes and the phenomenon position, at t=150, higher number of messages were sent to the sink than the messages at t=50, which may increase the end-to-end delay, thus number of retransmissions. Note that, in Fig. 5 sent and received packets includes the retransmitted and coordination packets.

VI. CONCLUSIONS

In this work, we address the reliability problem by defining *event reliability* and *query reliability* to reduce extra cost of reliable transport services, and to guarantee the minimal reliability to deliver all events and queries demanded by WSN applications. For this purpose, an energy-aware network topology is constructed using a distributed algorithm. Proposed scheme operates on static, randomly deployed network where nodes are location-unaware. Simulation results show that, significant savings on communication costs for event and query reliability are attainable while minimizing packet loss, energy consumption, and end-to-end delay. Our approach can be used by a wide range of WSN applications (e.g., fire monitoring, border surveillance, etc.) that require reliable communication service, energy efficiency, scalability, prolonged network lifetime and energy balance.

REFERENCES

[1] G. Holland and N. Vaidya. Analysis of TCP Performance over Mobile Ad Hoc Networks. In *Proc. of ACM Mobicom*, pages 219–230, 1999.

[2] Q. Li and D. Rus. Global Clock Synchronization in Sensor Networks. In *Proc. of IEEE Infocom*, 2004.

[3] Ns-2. http://www.isi.edu/nsnam/ns. 2004.

[4] S. Park, R. Vedantham, R. Sivakumar, and I. Akyildiz. A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks. In *Proc. of ACM MobiHoc*, pages 78–89, 2004.

[5] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In *Proc. of ACM MobiHoc*, pages 177–188, June 2003.

[6] F. Stann and J. Heideman. RMST: Reliable Data Transport in Sensor Networks. In *Proc. of the First IEEE Int. Workshop on Sensor Network Protocols and Applications*, pages 102–113, May 2003.

[7] N. Tezcan and W. Wang. TTS: Two-tiered Scheduling Algorithm for Wireless Sensor Networks. In *Proc. of IEEE ICC '06*, 2006.

[8] V. Tolety. Load Reduction in Ad Hoc Networks Using Mobile Servers. *Master Thesis, Colorado School of Mines*, 1999.

[9] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks. In *Proc. of ACM Int. Workshop on Wireless Sensor Networks and App.*, pages 1–11, 2002.

[10] F. Wang and Y. Zhang. Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response. In *Proc. of ACM Mobihoc*, pages 217–225, Lausanne, Switzerland, June 2002.

[11] Y. Wang, W. Wang, and X.-Y. Li. Distributed low-cost backbone formation for wireless ad hoc networks. In *Proc. of ACM MobiHoc*, pages 2–13, 2005.

[12] O. Younis and S. Fahmy. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, Oct-Dec 2004.

[13] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proc. of MobiSys*, pages 125–138, New York, NY, USA, 2004. ACM Press.