ART: An Asymmetric and Reliable Transport Mechanism for Wireless Sensor Networks

Nurcan Tezcan and
Wenye Wang
Department of Electrical and Computer Engineering,
North Carolina State University, Raleigh, NC 27695
E-mail: ntezcan, wwang@ncsu.edu

Abstract:
Many applications developed for wireless sensor networks (WSNs) demand for Reliable communication service, since majority of these applications are event-critical applications. There has been a vast body of knowledge on reliable data transfer in wireless networks; however, many of those solutions are not applicable to WSNs due to the fact that they address the problem by offering per message transport reliability. However, densely deployed sensor nodes can generate many redundant messages that essentially indicate the same *event* from the area of interest, this message-level reliability usually poses significantly high and unnecessary communication costs.

In this paper, we address the problem of reliable data transferring by first defining *event reliability* and *query reliability* to match the unique characteristics of WSNs. Unlike other studies on transport protocols for WSN, we consider event delivery in conjunction with query delivery. For the purpose, we propose an *energy-aware sensor classification* algorithm to construct a network topology that is composed of sensors in providing desired level of event and query reliability. Using such an approach, reliability is granted in the sense that critical event reports are received by the sink and queries are received by a small subset of sensors that can monitor the entire sensing field. We analyze our approach by taking asymmetric traffic characteristics into account and incorporating a distributed congestion control mechanism. We evaluate the performance of the proposed approach through an ns-2 based simulation and show that significant savings on communication costs are attainable while achieving event and query reliability.

Keywords: Reliable Query and Event Transport, Algorithm Design, Energy Conservation.

Reference to this paper should be made as follows: Tezcan, N., Wang, W. (2006) 'ART: An Asymmetric and Reliable Transport Mechanism for Wireless Sensor Networks', Int. Journal of Sensor Networks, Special Issue on Theoretical and Algorithmic Aspects in Sensor Networks.

Biographical notes: Nurcan Tezcan received the B.S. and M.S. degrees in computer engineering from Yeditepe University, Turkey in 2001 and Bogazici University, Turkey, in 2004, respectively. She is currently working toward the Ph.D. degree in computer engineering at the North Carolina State University, Raleigh. Her current research interests are in transport layer protocols, energy-efficient algorithms and performance analysis of wireless networks.

Wenye Wang received the B.S. and M.S. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1986 and 1991, respectively. She also received the M.S.E.E. and Ph.D. degree from Georgia Institute of Technology, Atlanta, Georgia in 1999 and 2002, respectively. She is now an Assistant Professor with the Department of Electrical and Computer Engineering, North Carolina State University. Her research interests are in mobile and secure computing, quality-of-service (QoS) sensitive networking protocols in single- and multi-hop networks.

*This work is supported by the National Science Foundation under Grant NO. 0524519.

1 INTRODUCTION

Wireless sensor networks (WSNs) are recently getting more powerful in sensing, collecting and disseminating information about the physical world. As they gain more capability on reporting important perceptions in a collaborative fashion, more and more applications will be developed for WSNs such as country border security, early fire detection etc., which are driven by the queries from the sink. Therefore, reliability of such mission critical applications or the success of these missions 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 on WSN?"

Consider an example of WSN applications for border surveillance. Many sensor nodes are scattered through a restricted area near a national border to monitor illegal border-crossing activity. Intruders in the area are detected; sensors report them immediately via event messages. Also, a centralized authority (through sink node) may further query the sensors for an up-to-date reading of their measurements, or update them to change detection parameters. In this example, each *event* such as border-crossing, must be reported successfully, but not necessarily every *message*. Further, every message from the sink must be reliably delivered to the entire sensing field, again not necessarily every node to achieve reliable information delivery between sensors and the sink.

In WSNs, a reliable delivery mechanism must provide reliability by handling the least possible number of messages in order to achieve significant energy conservation and low delivery latency under congested network conditions. Thus, we carefully define *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 requires an *energy-aware classification* of sensors through which a group of sensors, called *essential nodes* is selected to cover the entire sensible terrain. As a result of using a weighted-greedy algorithm, a *classified topology* is formed with those essential nodes that would communicate with the sink and participate in lost message recovery. Benefits of selecting such a group are twofold. First, end-to-end communications can take place between essential sensors and the sink which is well-fit to reliability requirements of WSNs. Thus end-to-end communication overhead can be reduced while achieving reliability. Second, having such a group allows easy incorporation of distributed congestion control mechanisms, which regulates the traffic flow by decreasing the number of active non-essential sensors if necessary. Built upon the classified topology, we propose an *Asymmetric Reliable Transport (ART)* mechanism that provides event and query reliability, dealing with only a subset of the large number of sensor nodes thus allowing a reduction in complexity in cases of loss message recovery. To adapt to the inherent characteristics of upstream (sensors-to-sink) and downstream (sink-to-sensors) traffic, we propose an asymmetric method, making use of ACK and NACK mechanisms.

There are a few transport layer mechanisms proposed for sensor networks [10, 12, 14, 18]. To the best of our knowledge, this work is the first attempt to make use of the classification approach to achieve event and query reliability. Considering reliability for upstream only [12, 14] or downstream only [10, 18] is not sufficient and it will restrict the potentials of a network protocol because reliable transmission service is fundamental to both sensor node and sink. Hence, two-way reliability feature should be available in a single transport mechanism.

The remainder of the paper is organized as follows. In Section 2, we present the network description and concept of reliability. We introduce a new classification algorithm in detail in Section 3. In Section 4, we present the design of reliability and congestion control schemes for event and query delivery. Performance evaluation is discussed in Section 5. In Section 6, existing transport solutions for WSNs are summarized, and Section 7 concludes the paper.

2 DEFINITIONS

2.1 Network Description

Let $S = \{s_1, s_2, s_3, \ldots, s_N\}$ be the finite set of sensors which are distributed randomly in a two-dimensional area $\mathbb{A}$. Each sensor s_i has a unique *id* (such as MAC address). We assume that each node is equipped to gather its location information via any lightweight localization technique for wireless networks [5]. Therefore, all sensor nodes and the sink know their location coordinates (x_i, y_i) and *sensing range* r_i. We assume that all nodes have similar processing and communication capabilities. Messages are sent in a multi-hop fashion.

The *sensing region* R_i of a node s_i is the area with its center at (x_i, y_i) and radius of r_i. A subset of sensors, $\mathbf{C} \subseteq S$ is called a *coverage set* if the union of the sensing regions of the $s_i \in \mathbf{C}$ covers the entire field $\mathbb{A}$ such that $\mathbb{A} \subseteq \bigcup_{s_i \in \mathbf{C}} R_i$.

The sensors are classified into essential (E) nodes and non-essential (N) nodes (more details about the classification algorithm is described in Section 3). This classification process is proceeded by finding a *coverage set*, denoted by $\mathbf{C}$. Let us denote the cardinality of coverage set $\mathbf{C}$ as $\mathcal{N}$ that is, $\mathcal{N} = ||\mathbf{C}||$. We consider a sensor node as an *essential* (E) node in $\mathbf{C}$ if $s_i \in \mathbf{C}$ and it is denoted as $s_i^{(E)}$;

otherwise, it is an non-essential (N) node, $s_i^{(N)}$. At any time, a unique coverage set is selected using a weighted-greedy algorithm explained in Section 3. The coverage set is valid for a time interval called *update interval*, denoted by $T_{\Delta U}$. In other words coverage set is determined periodically for every $T_{\Delta U}$.

Also, we assume that sensors are able to monitor their residual energy because many electronic devices are equipped with energy monitoring functions. The energy level of sensor s_i at the beginning of γ^{th} $T_{\Delta U}$, denoted by $e_i(\gamma \cdot T_{\Delta U})$, is calculated as:

$$e_i(\gamma \cdot T_{\Delta U}) = \frac{E_i(\gamma \cdot T_{\Delta U})}{E_i(0)}, \qquad (1)$$

where $E_i(0)$ is the initial energy corresponding to a fully charged battery [2], and $E_i(\gamma \cdot T_{\Delta U})$ is the residual energy of sensor s_i at the beginning of γ^{th} update interval. Hence, $e_i(\gamma \cdot T_{\Delta U}) = 1$ and $e_i(k \cdot T_{\Delta U}) = 0$ correspond to full and empty battery respectively.

In this context, a wireless sensor network is modeled as a directed graph $G(S, E)$, where S is the set of vertices ($||S|| = N$), representing the sensor nodes, and E is the set of edges, representing the communications links. We also consider the fact that links may be asymmetric due to radio irregularity [23]. A communication link is symmetric if there exists links from v_i to v_j and v_j to v_i, which is determined by using the *neighbor discovery* scheme given in [23].

2.2 Energy Model

The energy model of sensors is a function of reception energy consumption per bit ε_r and the transmission energy consumption per bit ε_t [8]. If node s_i sends a data packet of length l bits, an amount of $l \cdot \varepsilon_t$ energy will be deducted from sensors' residual energy, E_i. Let Ω^{up} and Ω^{down} be the energy consumed in upstream (sensors-to-sink) and downstream (sink-to-sensors) directions, respectively. Then

$$\Omega^{up} = l_u \cdot [N_t \cdot \varepsilon_t + c \cdot N_l \cdot (\varepsilon_r + \varepsilon_t)] \quad \text{and}$$
$$\Omega^{down} = l_d \cdot [N_r \cdot \varepsilon_r + (1 - c) \cdot N_l \cdot (\varepsilon_r + \varepsilon_t)],$$

where N_t, N_r and N_l are the numbers of transmitted, received, and relayed packets during one update interval $T_{\Delta U}$ on node s_i, respectively; l_u and l_d are the average lengths of upstream and downstream messages, respectively; and c is the ratio of relayed upstream messages over all relayed messages. Hence, residual energy of a sensor s_i at the beginning of the γ^{th} interval can be written as:

$$E_i(\gamma \cdot T_{\Delta U}) = E_i((\gamma - 1) \cdot T_{\Delta U}) - \Omega^{up} - \Omega^{down}, \qquad (2)$$

where $E_i((\gamma - 1) \cdot T_{\Delta U})$ is the residual energy at the beginning of the previous update interval.

2.3 Reliability Definitions

WSNs distinguish themselves from other wireless networks through traffic characteristics, e.g., asymmetric data traffic from sensors-to-sink and sink-to-sensors. Reliability of such networks are 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 downstream and upstream data delivery.

Consider a group of sensors need to send a sequence of messages to the sink node, s_o, regarding an event. *End-to-end reliable event transfer* is achieved when the first message indicating the event (sent by essential nodes) 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. However, subsequent messages regarding the same event does not affect event reliability.

Let v_k be the first message that reports event k to the sink. Then, the probability of successful transfer of an event k is given as follows:

$$Pr(success\ of\ v_k) = 1 - \prod_{s_i^{(E)} \in \mathbf{C}'} Pr\{\chi(s_i^{(E)}, s_o) = 0\}, \qquad (3)$$

where $\mathbf{C}' \subseteq \mathbf{C}$ is the set of essential nodes having sensed the event k. $\chi(s_i, s_o) \in$ [0,1] is a link state indicator function; $\chi(s_i, s_o) = 1$ indicates a link between s_i and s_o is up and enables communication, and $\chi(s_i, s_o) = 0$ indicates a down link. Note that, $\chi(s_i, s_o)$ is computed using independent failure probabilities of all links between s_i and s_o and it is a function of the physical medium and the underlying link layer protocols in use.

Consider K events occur in an update interval and they have to be delivered reliably. Then the expected number of successfully delivered events is $\sum_1^K Pr\{success\ of\ v_k\}$. Based on this expected number of successfully delivered events, we define *event reliability* metric to be the ratio of successful delivered messages such that:

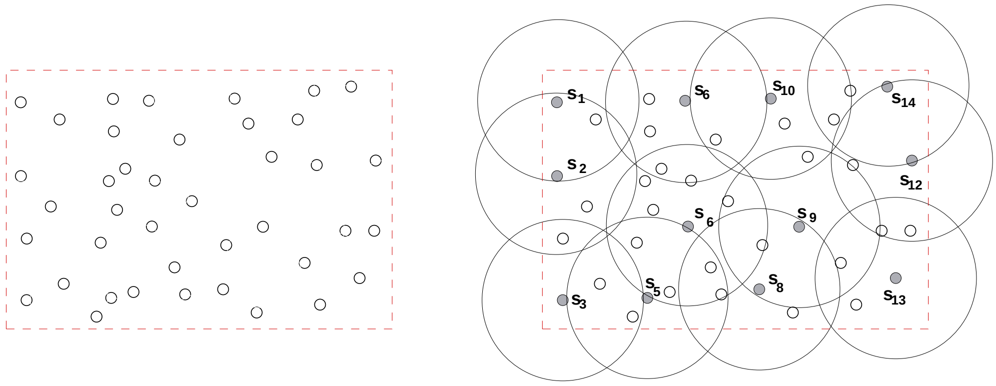
$$\mathcal{R}(v) = \frac{1}{K} \cdot \sum_{k=1}^{K} Pr\{success\ of\ v_k\}. \qquad (4)$$

Similarly, sink node has a sequence of queries, $[q_1, \ldots, q_k, q_{k+1}, \ldots, q_{K'}]$, which are sent to the sensor nodes. Then the *End-to-end reliable query transfer* is referred to as all queries are received by essential nodes successfully. The probability of the successful transfer of query k as:

$$Pr(success\ of\ q_k) = 1 - \prod_{s_i^{(E)} \in \mathbf{C}} Pr\{\chi(s_o, s_i^{(E)}) = 0\}. \qquad (5)$$

Note that we use only essential nodes in calculating $Pr(success\ of\ q_k)$ because sending query to essential nodes is sufficient to process the query in the entire field. If there is a number of K' queries to be sent during $T_{\Delta U}$. Then *query reliability* in an update interval, denoted by $\mathcal{R}(q)$, is defined as:

Figure 1: Classification of sensor nodes in ART.



(a) An example randomly deployed WSN. (b) Classification of example WSN.

$$\mathcal{R}(q) = \frac{1}{K'} \cdot \sum_{k=1}^{K'} Pr\{success\ of\ q_k\}. \quad (6)$$

Given event and query reliability definitions, we propose the new transport protocol, ART, to achieve 100% query and event reliability. Next, we will explain the sensor classification algorithm and reliability mechanisms of ART, respectively.

3 ENERGY-AWARE SENSOR CLASSIFICATION

The reliability of ART is built upon the classification of sensors as essential (E) nodes and non-essential (N) nodes. We propose to select the E-nodes by using a periodic weighted greedy algorithm running on the sink based on residual energy of sensors. For each update, nodes having higher energy levels are selected as essential to achieve fair energy consumption among sensors.

In order to select the set of E-nodes, we maintain a *coverage set*, denoted by **C**, to which E-nodes belong. The challenges involved in this process are (i) how can the coverage set be chosen? (ii) how can the coverage set be updated in order to maintain event and query reliability? In addition, we need to discuss whether a sink-based approach is a practical solution.

3.1 Node Classification Algorithm

For the first challenge, an ideal solution would be to find the *minimum* number of sensors that cover the entire field. However, it is an NP-hard problem similar to the well-known set cover problem. The goal in set cover problem is to cover a set with the smallest possible number of subsets given a ground set of elements [1, 13]. Due to this reason, we use a greedy approach to find an *approximating coverage set* running in polynomial time.

For different purposes, previous studies focused on the problem of finding near-optimal coverage in WSNs [3, 4, 20] . In [4], a greedy approach is proposed to find a connected set of sensors whose sensing regions cover an entire field. Therefore, a near-optimal coverage set is selected to form a connected network. However, our approach is different in two aspects. First, we do not need a connected set, since N-nodes can still be used to forward packets. Second, we choose the coverage set of sensors to maximize the *benefit* in terms of coverage, i.e., the largest uncovered sensing region is covered with the least sensors. As a result, our approach is to cover the entire field with minimum number of sensors having maximum residual energy.

Figure 1(a) shows an example sensor network where sensors are deployed randomly on a rectangular area $\mathbb{A}$. E-nodes and N-nodes are illustrated in different formatted circular dots, i.e., E-nodes with dark circles in Figure 1(b). Sensing region boundary of an E-node is plotted with dashed-circles. The union of sensing regions covers the entire sensing field. Therefore, by selecting the E-nodes, we guarantee that (i) when an event occurs, it is detected by at least one E-node and (ii) when the sink sends a query to all E-nodes, the query affects the entire sensing field.

We propose a *energy-aware greedy algorithm* to find a near optimal coverage set, given in **Algorithm 1**. In each step, **Algorithm 1** selects one node from the unselected sensors which covers the largest area with highest residual energy level. For this purpose, *weight function* is defined to represent the weight of a sensing region of a sensor based on its residual energy. For a given region, the weight based on the residual energy level of a sensor is:

$$w(i,\ R_i) = e_i \cdot [|R_i|], \quad (7)$$

where e_i is the energy level given in (1) and $[|R_i|]$ is the area of sensing region R_i.

Then, we calculate the *benefit* of selecting each sensor using the weight function. To do this, we first find the size of the area that can be covered by sensor s_i and has

Algorithm 1 Selecting Essential Nodes

Input: $S = \{s_1, s_2, s_3, \ldots, s_N\}$ is the set of sensors which are distributed randomly on $\mathbb{A}$.
A sensor has $s_i = (r_i,\, R_i,\, e_i,\, (x_i, y_i))$ where
r_i : the sensing range,
R_i : the sensing region,
e_i : residual energy level,
(x_i, y_i) : location coordinates.

Output: Coverage set, $\mathbf{C}$.

I.Initialize
  $\mathbf{C} := \emptyset$
  Let $R_\mathbf{C}$ be total sensing region of $\mathbf{C}$

II.Repeat
  Let $S - \mathbf{C} = \{s_1, s_2, \ldots, s_n\}$ be the candidates,
  max_benefit := 0;
  for each $s_i \in S - \mathbf{C}$
    Calculate the energy-benefit of s_i
    $benefit(s_i) := \sum_{a_j \in (R_i \cap \mathbb{A})/R_\mathbf{C}} w_i(a_j)$;
    if ($e_{benefit} \geq$ max_benefit)
      max_benefit := $benefit$;
      $temp := s_i$;
    end if;
  end for;
  $\mathbf{C} := \mathbf{C} \cup temp$;
Until $\mathbb{A} \subseteq R_\mathbf{C}$

III.Finalize
  Return $\mathbf{C}$;

not been covered yet. Consider the sensor s_i with sensing region R_i. Let $R_\mathbf{C}$ be the area that sensors of $\mathbf{C}$ covered so far, i.e., $\bigcup_{s_j \in \mathbf{C}} R_j$. *Beneficial area* of s_i is defined to be the region inside the sensing field which has not been covered, i.e., $R_\mathbf{C} = R_i \cap \mathbb{A})/R_\mathbf{C}$. Hence, benefit function for sensor s_i is the total weight of its beneficial area, which is given as:

$$benefit(s_i) = w(i,\ (R_i \cap \mathbb{A})/R_\mathbf{C}), \tag{8}$$

where R_i is the sensing region of sensor s_i and $R_\mathbf{C}$ is the total region covered by the sensors in $\mathbf{C}$.

Since **Algorithm 1** is to find a near-optimal coverage set, let us take a look how the proposed algorithm approximates an optimal coverage set.

Lemma 1. *Algorithm 1 gives a coverage set where the total weight of the entire field is $O(ln(N))$-factor of the optimal solution, where N is the number of sensor nodes.*

Proof. Let τ be the unit area and $\mathcal{A}$ be the size of the sensing field in terms of unit τ. **Algorithm 1** terminates when the sensing area of size $\mathcal{A}$ is fully covered. Consider the worst case where all N nodes have the minimum overlapping sensing regions covering the field, then all nodes will be selected as E-nodes.

Let each unit area have a *price* defined as follows:

$$price(\tau) = \{e_i \mid \tau \in R_i,\ s_i \in \mathbf{C}\}.$$

Algorithm 1 attempts to cover the entire field by maximizing the total weight, which is also equal to the summation of the price of each unit area in the sensing field, i.e., $\sum_{j=1}^{\mathcal{A}} price(\tau_j)$. At the j^{th} iteration, the remaining uncovered area can be covered by a total weight of at most $\frac{OPT}{\mathcal{A}-j+1}$, where OPT is the total weight of the optimal solution. Then we can write:

$$\textstyle\sum_{j=1}^{\mathcal{A}} price(\tau_j) \leq \sum_{j=1}^{\mathcal{A}} \frac{OPT}{\mathcal{A}-j+1} = OPT.H_\mathcal{A},$$

where $H_\mathcal{A}$ is harmonic number. Therefore, **Algorithm 1** finds an E-node set that covers the entire field at the cost of $O(ln\mathcal{A})$-*factor* of the optimal solution.

Consider a network with a total number of N sensors with sensing ranges r. When the sensors are placed such that overlapping sensing areas are minimum, size of sensing field will be at most $\sqrt{27}N(r)^2/2$ under the assumption of fully coverage [22]. Thus, the factor of the optimal total weight is obtained as $O(ln(N))$ for fixed sensing ranges. A loose bound of the running time of **Algorithm 1** is polynomial with upper bound $O(N^2)$. □

Finally, in Figure 2, we give an example showing how **Algorithm 1** finds the coverage set. Figure 2 (a) shows an intermediate step of the algorithm while Figure 2 (b) depicts the final status. In the first step, all nodes are candidates and the coverage set $\mathbf{C}$ is empty. Then, each run of Part II in **Algorithm 1** chooses an unselected node that has the maximum benefit. Figure 2 (a) shows the sensing field after the fourth run of Part II. In this example, sensor s_9, s_6, s_2 and s_{10} are selected based on their benefits and added to set $\mathbf{C}$. In the next step, uncovered area is $\mathbf{A}/\{R_9 \cup R_6 \cup R_2 \cup R_{10}\}$. In Figure 2 (a), we show the covered area, $R_\mathbf{C}$, in dark. Then, sensor having the benefit will be selected in the next step until the entire sensing field is fully covered as shown in Figure 2(b).

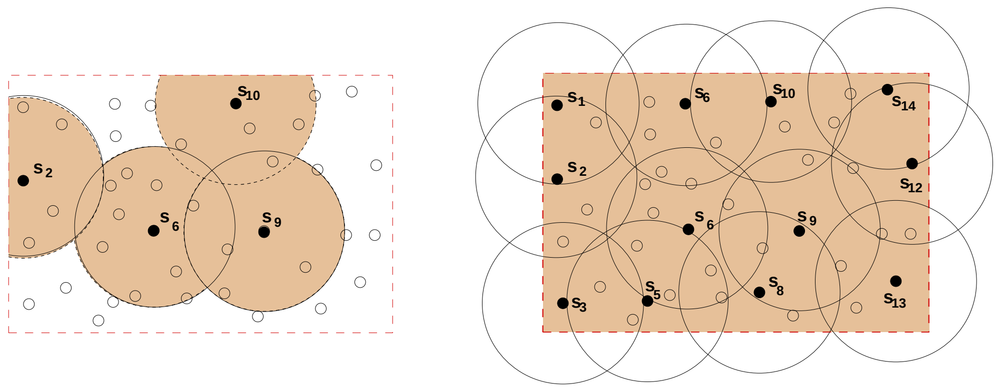
3.2 Update Coverage Set

The second challenge is how to update the coverage set. E-nodes should be updated throughout the lifetime of the WSN for two reasons: (i) to handle the unexpected E-node failures, (ii) to acquire fairly distributed energy consumption among sensors.

In general, there are two methods to updating the coverage set. The first method is called *global update* where all E-nodes are re-selected independent from the current set. In particular, global update is the process of repeating classification algorithm with latest residual energy levels of sensors. By this way, sensors that have overlapping regions and was E-Nodes in the previous round might be an N-node in the next update because more energy has been consumed when they were E-node before. This is used to acquire fairly distributed energy consumption among sensors. However, such a global maintenance may incur high overhead if repeated in short periods and can not handle the unexpected E-node failures during an update interval.

The second method is on-demand *local* update which can handle E-node failures immediately. Local update is

Figure 2: Walking-Through **Algorithm 1**.



(a) (b)

triggered when an unexpected E-node failure is detected by the sink. In this case, N-nodes covering the sensing region of failed E-node are assigned to be an E-Node by the sink. It may not be possible to find one N-node instead of failed E-node; however, it is much efficient instead of global update in any E-node failure.

In ART, we combine local and global such that, in case of an E-node failure new E-nodes are selected locally where global update will be performed for longer predetermined update intervals. Sink can monitor up-to-date energy reserves of sensors using a *energy monitoring* scheme [21]. Based on this remaining energy of sensors, a new essential set is formed by running **Algorithm 1**. After each global update, sink informs sensors of their type by using a control message. The effects of update interval on network lifetime and energy consumption is discussed in Section 5.

3.3 Discussion

Now we elaborate on the reasonings behind proposing a sink-based or a centralized algorithm for classification. The main reason behind the centralized approach is the residual energy information of a node on which node selection algorithms are based. For a centralized approach, global information about the state of a sensor network together with node coordinate information need to be disseminated within the network towards a single node (e.g., sink node). In [1], it is shown that collecting information from a sink node is more power-efficient manner compared to spreading this information to each and every other node within the network. In addition, choosing the sink node as the target of data propagation is reasonable if we considers that the sink node has ample energy and computing power compared to individual sensor nodes. Having the global view of the network at the sink node provisions algorithms for closer-to-optimal coverage set determination as well.

Finally, using a centralized scheme can relieve processing load from the sensors in the field and help in extending the overall network lifetime by reducing energy consumption at individual nodes. The proposed greedy algorithm runs on the sink with an approximation ratio of lnN, providing very close-to-optimal coverage sets for most instances of the sensor deployments. Additionally, maintaining the node set selections (i.e., E-node updates) can be realized through low cost information diffusion methods.

4 ART OPERATIONS

ART is an *asymmetric and reliable transport* mechanism which provides end-to-end reliability in two directions based on energy-aware node classification and a congestion control mechanism. In this section, we describe the details of ART protocol operations, which includes three main functions:

1. Reliable query transfer
2. Reliable event transfer
3. Distributed congestion control

After classifying sensors as essential and non-essential sensors, end-to-end reliable communications are provided by using asymmetric acknowledgement (ACK) and negative acknowledgement (NACK) signaling between E-nodes and the sink node. Then, we propose a distributed energy-aware congestion control mechanism which relies on receiving ACK packets from the sink. When congestion is detected, ART simply regulates data traffic by temporarily squelching the traffic of N-nodes. Note that, when there is no congestion, both E-nodes and N-nodes participate in relaying messages to the sink. However, only E-nodes are responsible in providing end-to-end event and query reliability by recovering the lost messages.

4.1 Reliable Query Transfer

Reliable query (sink-to-sensors) transfer is provided using negative acknowledgements sent from E-Nodes to the sink

Algorithm 2 Reliable Query Transfer

Input: Given a sensor network G; Sink has a set of queries Q = $[q_1, \ldots, q_k, q_{k+1}, \ldots]$.

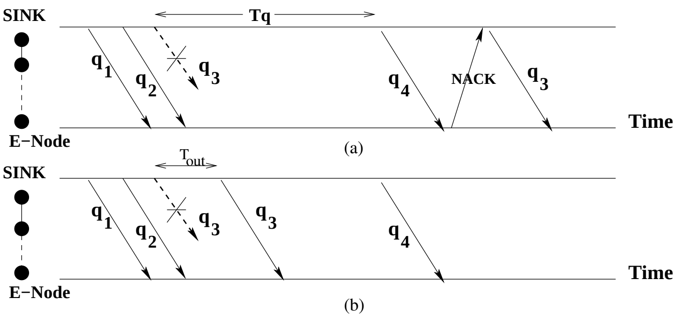
1. *Sink:* Send the in-sequence queries with sequence numbers 1,2,..k.
2. *E-Node:* Receive the messages for q_k. Check the sequence number for loss detection.
3. *E-Node:* If a gap is detected in the sequence numbers, send an NACK to recover the lost message.
4. *Sink:* Retransmit q_{k-1} if a NACK is received.
5. *E-Node:* When the queries are successfully received, check P/F bit. If P/F bit is set, send an ACK to the sink. (details in Section 4.3)
Sink: Retransmits the message with P/F bit is set until the ACK is received.

if there is a query loss. Since the queries sent by the sink are in order, sensors can detect the lost message by use of *sequence numbers* in the query messages. An NACK message is sent if a gap is detected, i.e., an out of sequence number, when sink sends a new query message to the E-Nodes. When an E-Node detects a gap in the sequence number of the new query, it sends an NACK back to the sink to recover the previous query. This procedure is described in **Algorithm 2**.

However, lost query messages can be detected when E-Nodes receive a new query message. This may result in two problems. First, loss of the last query message can not be detected. Consider the last message q_k with sequence number k is lost. E-Node may not handle the lost message since there is no consecutive query. Second, the query transmission frequency might be very low such that lost queries can not be recovered before timeout. 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 not be sent before timeout. And the sink retransmits this message until an ACK is received because ACK mechanism is used in reliable event transfer (see details in Section 4.2). Therefore, E-Nodes which receive a query with P/F bit set send an ACK to the sink, indicating the query is received successfully.

An example query transmission scenario is illustrated in Figure 3. In Figure 3 (a), the P/F bit is not used. When the sink sends queries 1, 2 and 3 consecutively where query 3 is lost. After query 3, the sink decreases the query transmission frequency and sends q_4 after a time period T_q. In this case, q_3 is recovered when q_4 is received. If loss recovery period (i.e., T_q) is very long, even though q_3 can be recovered, long recovery period may affect the performance. Instead, the same scenario is depicted when P/F bit is set in Figure 3 (b), where q_3 is recovered before the next query since an ACK is not received at the sink. This method is very helpful when the query traffic pattern is not uniformly distributed in which case, the interarrival time between queries are not constant. Then the use of P/F bit makes the transport protocol flexible and reliable.

Figure 3: Example of Query Loss.



4.2 Reliable Event Transfer

The NACK mechanism used in *query transfer* does not work for reliable event transfer 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.

For event reliability, we propose a *lightly-loaded ACK* mechanism between the E-Nodes and the sink node given in **Algorithm 3**. Each E-Node waits for acknowledgement for only the first message that reports an event, i.e., *event-alarm.* When a new sensing value is obtained, an E-Node 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. EN bit, is used to force the sink to send acknowledgement. Event-alarm rate depends on the distribution of events detected in the sensing field. Similar to downstream communications, only the E-Nodes are responsible for waiting the acknowledgement and may retransmit if necessary.

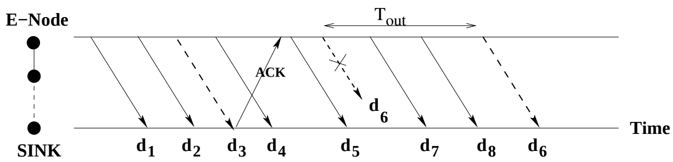
As an example, in Figure 4 an event transfer scenario is illustrated where v_3 and v_6 are event-alarm messages and their EN bits are set. In this example, the first event alarm message is received by the sink, and the ACK is transmitted. However, next alarm message v_6 is lost. Since the sender is responsible for loss detection and recovery, E-Node retransmits v_6 after retransmission timeouts shown

Algorithm 3 Reliable Event Transfer

Input: Given a sensor network G; An E-Node is sending an event-alarm v_k^{EA} given timeout t_{out};

1. *E-Node:* If $v_k = v_k^{EA}$, set the EN bit, send to the sink, then start timer and buffer v_k^{EA} until an ACK is received.
Otherwise, send it to the sink, delete from the buffer.
2. *Sink:* Send an ACK if it receives a v_k^{EA}
3. *E-Node:* If the ACK is not received for v_k^{EA} retransmit v_k^{EA} and reset timer.

Figure 4: Example of Event-Alarm Loss.



in Figure 4. Therefore, loss recovery is triggered only for event-alarm messages by the E-Node, which is very effective in energy saving as shown in Section 5.

4.3 Distributed Congestion Control

Given a WSN consisting of large number of nodes, congestion is an inevitable problem because a large number of sensors may transmit the sensed event at the same time. To detect and avoid congestion, several works has been proposed using different mechanisms in WSNs [7, 12, 17]. In [12], congestion is detected by monitoring the buffers of sensor nodes. When congestion is detected, sensors inform the sink node to decrease the reporting frequency of the network. A different method is that congestion detection is based on local channel monitoring and the congestion is propagated through hop-by-hop back-pressure messages upstream toward the source [17].

Unlike these existing solutions, in ART, congestion control is handled by the E-nodes in a distributed manner. It is based on monitoring the ACK packets of event reports. If an ACK is not received during a timeout period by the E-node, traffic of non-essential sensors is reduced by sending them congestion alarm messages, which will temporarily make them stop sending their measurements. When an ACK is received, congestion-safe message is announced to resume normal operation of the network.

It is possible that lost data packets may not indicate an accurate congestion for WSN because losses can be caused by link failures or congestion [15]. However, in ART we monitor only the event-alarm messages which report the sensed events. Congestion often occurs when events are reported by several sensors [17]. When an event is detected, many correlated event messages are sent to the sink, especially if the event is sensed in a large area and the network is dense, e.g., earthquake detection. Thus, monitoring the loss event-alarm messages is an efficient and simple way to detect congestion. Another advantage of this mechanism is its ease of use, since we already use a timer for retransmission of event-alarms. Thus, congestion timeout can be determined in accordance with retransmission timeout, which will be explained in Section 4.4. Each E-node decides and triggers the congestion control procedure without the centralized control of sink, based on receiving the ACK of an event-alarm.

In case of congestion timeout, E-nodes make their neighboring N-nodes temporarily *passive* via congestion alarm messages. Being passive for a sensor here means not sending sensing measurements to the sink. First, an E-node, which detects a congestion, will broadcast a *congestion alarm* (CA) message. After timeout period, if the congestion is still not relieved, an E-node will resend the CA by increasing the *hop-count*. This will continue until the congestion is relieved. From the N-nodes point of view, when they receive a CA message, they temporarily stop sending their sensing measurements and decrease the hop-count. The CA message is flooded until the hop-count is 0.

Note that N-nodes may receive multiple CA messages, $CA = [CA^1, CA^2, ..]$, which are sent by an E-node sequentially until the congestion is relieved. Each CA^j includes hop-count denoted by *hop-count(j) = j*. Because E-Nodes increase hop-count in every CA message, CA messages are flooded up to j-th neighbors of the E-node. When the ACK of an event is received, E-node sends *congestion safe* (CS) message similar the CA to resume the normal operation of the network. Congestion safe message is sent with the hop-count value of the latest CA message by the E-node. Therefore, the number of sensors sending their measurements is reduced, thus regulating the excessive traffic for congestion control.

4.4 Timeout and Retransmissions

In ART, we use an asymmetric protocol, e.g., NACK for sink-to-sensor and ACK for sensor-to-sink communication. While using NACKs, the sink only retransmits if it receives an NACK for a query message. Therefore, no timer is used. However, while transferring events from sensors to the sink, E-nodes wait ACKs for event-alarm messages. When an E-node sends an event alarm message, it triggers the timer and waits for *timeout period* to detect congestion or retransmit. Thus, timeout periods becomes particularly important and will be discussed in this section.

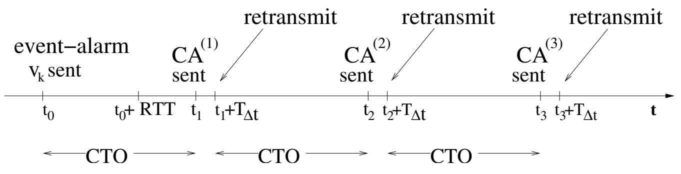
ART uses timeouts for both reliable end-to-end delivery and congestion control. We use *congestion timeout* (CTO) for congestion detection, which is dynamically determined based on *round trip time* (RTT) similar to adaptive retransmission timeout in TCP. Assume that all sensors have an initial RTT that 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, $RTT(sample)$ is computed dynamically based on the latest RTT by using the *time stamp* field. Sensors assert the time information in their messages sent back via ACKs by the sink. Thus, E-nodes can determine the $RTT(sample)$ by comparing the time stamp received by ACK. Then. the estimated RTT is determined by exponential averaging as:

$$RTT(t) = \alpha * RTT(t-1) + (1-\alpha) * RTT(sample)$$
$$CTO(t) = \eta * RTT(t),$$

where $\alpha \in (0,1)$ is weight ratio, and $\eta > 1$ indicates the coefficient of delay tolerance of the application.

Retransmission is done after each *retransmission timeout* (RTO) when necessary. However in ART, we let RTO equal to CTO used for congestion detection. CA messages

Figure 5: Retransmission and Congestion Control Behavior.



are sent when the timer expires. Clearly, it is not efficient to retransmit and send the CA message at the same time. Hence

$$RTO(t) = CTO(t) + \xi, \tag{9}$$

where ξ is the one-hop transmission delay. Event alarm messages are retransmitted after ξ of sending the CA message. Note that, different from other wireless and wired transport protocols, retransmission does not block the next data transmissions in WSNs. We continue sending next messages and retransmit the lost message if needed. The detailed time diagram of retransmission and congestion control in ART is shown in Figure 5 in which t is the time instant.

- $t = t_0$: Suppose an E-node detects an event at time t_0 and immediately reports it by setting the EN bit of the message. And CTO timer starts to count down.

- $t = t_1$: At time t_1, CTO timer expires. As shown in Figure 5, CTO is greater than the estimated RTT. When the timer expires, the first congestion alarm message is sent by the E-node. The N-nodes which receive the $CA^{(1)}$ do not send their measurements until they receive a congestion safe message.

- $t = t_1 + T_{\Delta t}$: According to **Algorithm 2**, E-node waits until $t = t_1 + T_{\Delta t}$. Then, it retransmits the event-alarm. By retransmissions, the CTO timer is restarted.

- $t = t_2$: During $t = t_1 + T_{\Delta t}$ to $t = t_2$, the E-node continues waiting for the ACK. At time t_2, since CTO timer expires, $CA^{(2)}$ is sent by increasing the hop-count. Until receiving an ACK from the sink, retransmission and CA steps will be repeated consecutively similar as $t = t_1$ and $t = t_1 + T_{\Delta t}$.

In ART, only the first message with an event information needs to be acknowledged and retransmitted if necessary. Since this message has a setting bit EN bit, the sink will send an ACK for this message. In other words, event information is guaranteed to reach the sink node, whereas not every message is guaranteed, which is to achieve the objective of *event reliability*. Therefore, the number of retransmissions is decreased which in turn will reduce the energy consumption.

5 PERFORMANCE EVALUATION

The proposed ART protocol is implemented in the ns-2 [9] network simulator. We conducted several simulations using different scenarios in a static sensor network. The performance of ART is evaluated regarding the effectiveness of classification algorithm, energy balance, network lifetime, and node density.

5.1 Performance Metrics

We use the following metrics to characterize the performance:

- *E-Node ratio*: It is the ratio of the number of E-nodes to the number of sensors in a sensing field. We use E-Node ratio to represent the effectiveness of node classification, i.e., size of coverage set, resulting from our weighted-greedy algorithm in Section 3.

- *Residual energy*: It is the amount of energy remaining in a sensor, which is measured at the beginning of every update interval.

- *Network lifetime*: It represents the maximum time interval that a network can maintain its functionality. We consider a WSN as alive when every point in **A** is covered by at least one sensor.

- *End-to-end delay*: It is the time for a packet to arrive at transport entity of the receiver after transmitted by the transport entity of the sender.

- *Packet loss ratio*: It is the ratio of the number of packets lost to the number of packets generated.

5.2 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 300 x 300 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. Note that only the density of sensors affects the performance of the node classification algorithm, thus there is no need to vary the size of the area.

In the basic scenario, 100 fixed sensor nodes having transmission range of 90 m and sensing ranges of 60 m are used. We use the energy model given in Section 2.2 where initial energy of sensors are 3 J.

Before we describe the performance results, we explain the application run on sensors and the sink. 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 [16] at a maximum

Table 1: Simulation Parameters

Area of sensing field	300 x 300 m^2
Number of sensor nodes	100
Radio range of a sensor node	90 m
Sensing range of a sensor node	60 m
Packet length	100 bytes
Interface Queue length	50
Transmit power	24 mW
Receive power	13 mW
Idle power	13 mW

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* 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*, as a simulation parameter to maintain traffic load in downstream direction. Queries sent by the sink do not affect the scenario or sensing period in the simulation. The coordinates of the sink is the center of the sensing field and same for all experiments. CSMA/CA is used as the MAC protocol and AODV is used as the routing protocol [11].

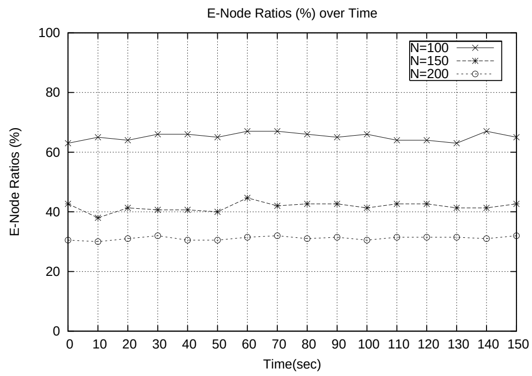
5.3 Simulation Results

We start by illustrating the effectiveness of the *energy-aware* node classification algorithm (**Algorithm 1**), i.e., the size of coverage set for various network densities. We then discuss the effect of update interval, which is an important question to find the value of *update interval* for a given sensor network. We then show the effect of update interval on residual energy and network lifetime. Further, performance gains of reliable event and query transfer service are shown over message-level reliable service. These gains demonstrate the unnecessary overhead that is generated when message-level reliability is concerned instead of event and query reliability which is guaranteed by persisted retransmissions for all experiments.

5.3.1 Performance of Node Classification Algorithm

Figure6 plots the ratio of E-nodes for different network densities. Note that, given the fixed area of sensing field, the network density depends on the number of nodes. Among the three, the network with higher density (200 nodes) has the lowest ratio of E-nodes, thus showing that

Figure 6: The E-Node Ratio of Coverage Set vs. Network Density ($T_{\Delta U} = 10$ sec).

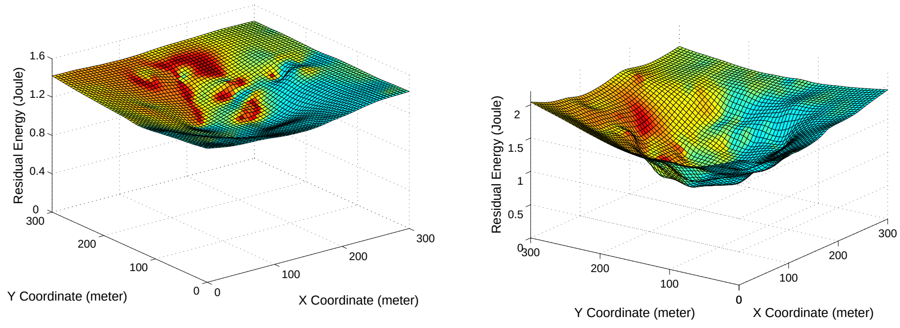


the greedy algorithm is able to find coverage sets regardless network density with a decreasing E-node ratio as the network density increases. Therefore,, the weighted-greed algorithm for node classification is even more effective in reducing the cost of reliability in dense networks. Further, results in Figure 6 indicate that the ratio of E-nodes does not vary in time. In every 10 seconds, our algorithm finds a new coverage set which is independent from the previous set. This implies that no matter what changes occurred in an update interval, E-nodes can always be selected.

The effect on energy balance of node classification algorithm is visually shown in Figure 7 which indicates the residual energy of sensors located randomly on the sensing field. The xy plane represents the sensing field and z coordinate is their residual energy. In Figure 7 (a), there are 100 nodes and the sink node is located at the center. A coverage set is selected at the beginning and updated periodically every 10 sec. Event-reporting frequency and query-reporting frequency are set to 0.5 sec and 5 sec., respectively. We repeated the same scenario with same parameters in 7 (b), except the coverage set is not updated.

Figure 7 (b) clearly illustrates that the sensor nodes around the sink (which is located at the center) consume much higher energy than the other nodes in the network, since they are used more frequently to relay the packets to the sink. The non-uniform residual energy distribution within the entire sensor network may lead to network partitioning and shorten the network lifetime. Severity of such non-uniform energy distribution is alleviated by updating the E-nodes as shown in Figure 7 (a). An advantage of **Algorithm 1** is that it adopts *benefit* function while updating the coverage set. This reduces the variance in sensors residual energy. However, the sink has to send a message to sensors after updating E-nodes. We see that the maximum residual energy in Figure 7 (a) is about 8% greater than the maximum residual energy in (b). In order to observe the effects of update interval, average residual energy will be discussed next.

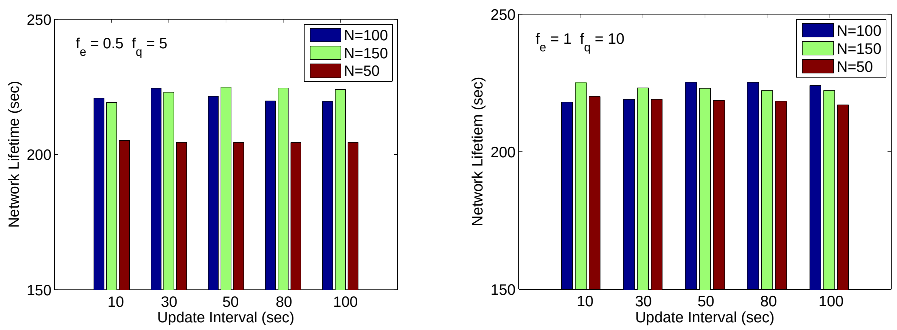
Figure 7: Residual Energy Distributions: N = 100 nodes at t = 150 sec.



(a) Coverage Set is Updated.

(b) Coverage Set is Not Updated.

Figure 8: Effect of Update Interval on Average Residual Energy and Lifetime.



(a) Lifetime vs. Update Interval of Heavy-loaded traffic.

(b) Lifetime vs. Update Interval of Lightly-loaded traffic.

5.3.2 Effect of Update Interval on Residual Energy and Lifetime

For performance evaluation of ART protocol, selecting a proper time interval is very important for a given network. A very large value may increase the communication cost, thus reducing residual energy. On the other hand, a very small update interval may cause high variance in sensors residual energy as E-nodes may drain out their battery much faster, thus partitioning the network. Thus, we study the effect of update interval on residual energy and network lifetime.

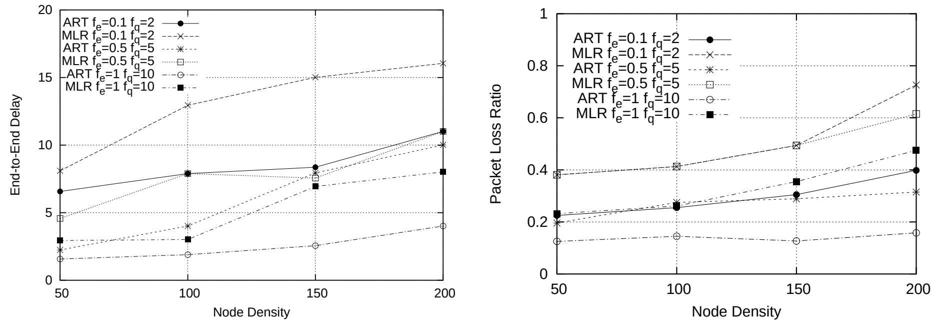
However, the optimal interval length should be the one that prolongs, if not maximize, network lifetime. In Figure 8 (a) and (b), we plot network lifetime against update interval for various dense networks. We observe that update interval changes the lifetime at most 3% as a result of energy consumption and residual energy distribution. For example, consider the network having 100 nodes. Lifetime curve has a peak point at interval length 30 for heavy loaded scenario in Figure 8 (a). Before and after 30, lifetime curve follows a non-decreasing and non-increasing trend, respectively. We observe the same trend in lightly loaded scenario of network with 100 nodes in Figure 8 (b). However, this time the peak value is 80 sec. This signifies that scenario with lightly-loaded traffic has longer update interval than the heavy traffic network.

In this experiment, we also present the performance of networks having different packet load. We generate different packet loads by varying the event-reporting (f_e) and query-reporting frequency (f_q) parameters. In Figure 8, two different packet loads are performed: (i) heavy: $f_e = 0.5$ and $f_q = 5$ and (ii) light: $f_e = 1$ and $f_q = 10$ sec.

5.3.3 Query and Event Reliability

The *ideal* reliable service for a WSN is achieved when the 100% query and event reliability is provided. Thus, retransmissions are needed only to recover the loss of event alarm messages and loss of query messages delivered to the E-nodes. We observe that for message-level reliable service, unnecessary overhead will be generated, which implies the ART protocol has gained performance improvement for ideal reliable service is not used. We refer message-level reliable service as *MLR* and compare the performance of our query and event reliability scheme which

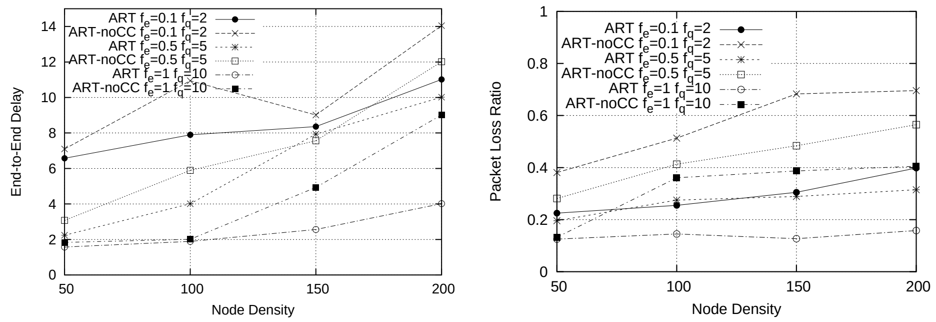
Figure 9: End-to-End (E2E) Delay and Packet Loss: ART and MLR.



(a) Effect of Query/Event Reliability on E2E Delay.

(b) Effect of Query/Event Reliability on Packet Loss.

Figure 10: Effect of Congestion Control Mechanism.



(a) Effect of Congestion Control on E2E Delay.

(b) Effect of Congestion Control on Packet Loss.

is referred as *ART* to *MLR* for various network densities and packet loads. Initial roundtrip time ($RTT = 2$ *sec*), coefficient of adaptive RTT ($\alpha = 0.125$), coefficient of delay tolerance ($\eta = 0.8$) and one-hop transmission delay ($\xi = 0.05$ *sec*) are used as input parameters.

Figure 9 compares the performance of ART and MLR with respect to average end-to-end delay and packet loss ratio. 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. From Figure 9 (a), we find that the end-to-end delay is a function of increasing network density. Notice that under all traffic loads, end-to-end delay of ART is 40% lower than MLR on average. The reasons for reduced delay are twofold: the advantage gained by having classified E-Nodes dealing with retransmissions reduces the amount of data sent, and the advantage gained by using event-based reliability to avoid ACK implosion. Also, end-to-end delay in ART degrades gracefully with decrease in traffic load. even in heavy packet load, the delay in ART protocol remains below 5 sec. The packet loss ratio is shown in Figure 9 (b) where even at heavy load, ART yields less packet losses than MLR scheme.

5.3.4 Effect of Congestion Control Mechanism

The congestion control scheme of ART is designed to reduce the effect of congestion, e.g., high packet loss and long delay. The distributed congestion control scheme described in Section 4.3 does not guarantee to detect each and every congestion; however, it is very effective in regulating traffic load and maintaining ideal reliability without additional overhead. We used the same type of traffic load scenarios as in Figure 9 to observe the effect of congestion control. In Figure 10 (a), we observe that end-to-end delay is significantly reduced as the density of the network increases. Moreover, packet loss is reduced up to 50% by using the congestion control mechanism. The reason is that, only E-nodes sensing the phenomena, which avoids the neighboring N-nodes that sense the same event to send their reports if necessary. Note that, in ART, we do not include a scheduling scheme for N-nodes, i.e., N-nodes do not turn off their radio. Even in a congestion alarm, N-nodes participate in relaying the messages to the sink. However, they do not send new sensing measurements, thus the number of messages injected to the network is reduced significantly compared to ART without congestion control.

Table 2: Comparison of Existing Transport Protocols

	PSFQ	*RMST*	*ESRT*	*GARUDA*	*ART*
Reliability	Downstream HopbyHop NACK	Upstream HopbyHop NACK	Upstream End2End -	Downstream HopbyHop NACK	Both End2End ACK/NACK
Energy-aware	-	-	Yes	-	Yes
Loss Rec.	Yes	Yes	-	Yes	Yes
Cong. control	No	No	Yes	No	Yes

6 RELATED WORK

The reliable transport problem in wireless packet-data networks has been studied in several research works many of which are aimed to improve the performance of TCP over wireless links and ad hoc networks [19]. In [15], a new transport protocol, *Ad hoc Transport Protocol*, is proposed for operating conditions in ad hoc networks. However, it is designed for point-to-point data transport for mobile nodes, thus consisting procedures such as connection initiation and rate based transmission which cannot be used in WSNs due to energy constraint of sensors.

Existing transport layer protocols designed for upstream or downstream reliability in WSNs are either sink-to-sensor [10, 18] or sensor-to-sink reliable delivery [6, 12, 14]. Table 2 shows a detailed comparison of their characteristics.

Pump Slowly, Fetch Quickly (PSFQ) [18] is the first transport protocol proposed for downstream reliable data transmission from source to the sensor nodes. This protocol is based on a set of operations including hop-by-hop error recovery, in-network caching and sending repair request via NACKs (Fetch) that is faster than the source transmission rate (Pump). Also, a hop-by-hop error recovery mechanism is used for message loss recovery. Although PSFQ achieves in-sequence transmissions, with specific reference to a re-tasking application, it cannot handle single packet losses, and it also does not consider losses due to the congestion. *GARUDA* [10] is another important work focusing on reliable downstream data delivery based on a virtual infrastructure, that is, a set of local and designated loss recovery servers. This solution also supports multiple reliable semantics such as delivery to sensors in a sub-region of the field. Fast loss recovery is down by using a two-phase loss recovery strategy: the first one involves the core nodes recovering from all lost packets, and then the recovery of lost packets at the non-core nodes.

From the upstream perspective, *Event-to-Sink Reliable Transport* (ESRT) [12] is the first protocol which is motivated by the fact that the sink is only interested in reliable detection of event features from the collective information provided by sensor nodes. Although it considers the event information, rather than each individual data packet, it is designed for sensor-to-sink communications only. The event-to-sink reliability is determined by the number of received data packets. ESRT adjusts the reporting frequency of the source nodes to increase and decrease the reliability. Further, ESRT regulates the reporting rate of sensors in response to congestion detection in the network by using congestion notification bit and reducing event reporting frequency. *Reliable Multi-Segment Transport* (RMST) [14] is another transport layer protocol which is designed to run in conjunction with directed diffusion. It is a selective NACK-based protocol, which is used for transfer large amount of data from sensors to the sink. The receiver sensors are responsible for detecting whether a fragment needs retransmission, thus achieving reliable data transfer.

To the best of our knowledge, ART is the first bidirectional transport protocol for reliable event and query transmission in WSNs . The proposed protocol addresses the reliability requirements for both sensor-to-sink and sink-to-sensor data transfer. In addition, incorporating congestion control mechanism shows considerable performance improvement in terms of energy savings and balancing, which further improves network lifetime.

7 CONCLUSION

In this paper, we introduced a new transport protocol addressing bidirectional end-to-end reliability in wireless sensor networks. First, we propose an *energy-aware* node classification algorithm, which is a weighted-greedy algorithm to select a set of sensors, called *essential nodes*. The algorithm takes into account the remaining battery power on the sensors so that sensors running low on battery has a smaller chance of being essential. This gives a significant flexibility for balancing the available energy in the network among all sensors, thus providing a longer network lifetime.

The reliable event and query transfer is accomplished between the sink and essential nodes, while incurring low overhead in terms of control messages and retransmissions. Second, for event transfer, a lightweight ACK mechanism is used while NACK solves the reliable query delivery. Third, we incorporated a distributed congestion control mechanism, in which congestion is relieved by regulating traffic from non-essential sensor nodes.

Simulation experiments have validated that, under the 100% reliable delivery between essential nodes and the sink, traffic load in the network is dramatically reduced by the integration of node classification and congestion control. Th proposed protocol performs significantly better than message-level reliability scheme in terms of latency and packet loss.

REFERENCES

[1] Z. Abrams, A. Goel, and S. Plotkin. Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks. In *Proc. of Int. Conference on Information Processing in Sensor Networks*, April 2004.

[2] J.-H. Chang and L. Tassiulas. Maximum Lifetime Routing in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(4):609–619, August 2004.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proc. of ACM Mobicom*, pages 85–96, July 2001.

[4] H. Gupta, S. R. Das, and Q. Gu. Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. In *Proc. of ACM Mobihoc*, June 2003.

[5] T. He, C. Huang, B. M. Blum, and J. A. Abdelzaher. Range-free Localization Schemes for Large Scale Sensor Networks. In *In Proc. of ACM Mobicom*, pages 81–95, September 2003.

[6] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *Proc. of ICDCS*, May 2003.

[7] B. Hull, K. Jamieson, and H. Balakrishnan. Techniques for Mitigating Congestion in Sensor Networks. In *Poc. of SenSys*, 2004.

[8] L. Lin, N. Shroff, and R. Srikant. A Distributed Power-aware Routing Algorithm with Logarithmic Competitive Ratio for Sensor Networks. *Technical Report Purdue University*, 2002.

[9] Ns-2. http://www.isi.edu/nsnam/ns. 2005.

[10] 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, May 2004.

[11] C. E. Perkins, E. M. Royer, , and S. R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561, IETF Mobile Ad Hoc Networks Working Group.*

[12] 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.

[13] P. Slavk. Energy-efficient Target Coverage in Wireless Sensor Networks. In *Proc. of ACM Symposium on Theory of Computing*, pages 435 – 441, 1996.

[14] F. Stann and J. Heideman. RMST: Reliable Data Transport in Sensor Networks. In *Proc. of IEEE SNPA*, pages 102–113, May 2003.

[15] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar. ATP: A Reliable Transport Protocol for Ad-hoc Networks. In *Proc. of ACM Mobihoc*, pages 64–75, June 2003.

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

[17] C.-Y. Wan and A. T. Campbell. CODA: Congestion Detection and Avoidance in Sensor Networks. In *In Proc. of the First ACM Conference on Embedded Networked Sensor Systems*, November 2003.

[18] 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 Applications*, pages 1–11, 2002.

[19] 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, June 2002.

[20] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *Proc. of SenSys*, pages 28–40, 2003.

[21] 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, May 2005.

[22] R. Williams. The Geometrical Foundation of Natural Structure: A Source Book of Design. In *Dover Pub. Inc*, pages 51 – 52, 1979.

[23] 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.