GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks

Seung-Jong Park, *Member*, *IEEE*, Ramanuja Vedantham, *Member*, *IEEE*, Raghupathy Sivakumar, *Senior Member*, *IEEE*, and Ian F. Akyildiz, *Fellow*, *IEEE*

Abstract—There exist several applications of sensor networks where the reliability of data delivery can be critical. Although the redundancy inherent in a sensor network might increase the degree of reliability, it by no means can provide any guaranteed reliability semantics. In this paper, we consider the problem of reliable sink-to-sensors data delivery. We first identify several fundamental challenges that need to be addressed and are unique to the environment of wireless sensor networks. We then propose a scalable framework for reliable downstream data delivery that is specifically designed to both address and leverage the characteristics of the wireless sensor networks while achieving the reliability in an efficient manner. Through ns2-based simulations, we evaluate the proposed framework.

Index Terms—Wireless sensor networks, reliable transport protocols.

1 Introduction

The increased awareness of the wide variety of applications for sensor networks has spurred a tremendous amount of research over the past few years [1]. Because of the frugal energy budget of sensor networks, a significant amount of such work focuses on energy-aware network protocols [2], [3], [4]. In addition to the energy conservation problem, sensor networks suffer from a high rate of data loss due to wireless channel errors, congestion, and broadcast storm [5]. Under the high rate of data losses, unreliable data deliveries increase the odds of data retransmission and, hence, waste a significant amount of valuable energy. Therefore, it is necessary to consider the robustness of protocols while taking into account energy conservation.

In this paper, we consider the problem of *reliable downstream point-to-multipoint data delivery*, from a sink to sensors, in wireless sensor networks (WSNs). The need for the reliability in WSNs is dependent on the type of applications. Consider a security application where image sensors are required to detect and identify the presence of critical targets. Given the critical nature of the application, it can be argued that any message from the sink has to reach the sensors reliably.

Now, for this security application, the sink may send one of the following three classes of messages, all of which have to be delivered reliably to the sensors: 1) If the underlying network is composed of reconfigurable sensors that can be reprogrammed [6], the sink may want to send a particular (say, upgraded) image detection/processing software to the sensors. We refer to such messages as the *control code.* 2) Next, the sink may have to send a database of target images to the sensors to help in the image recognition triggered by subsequent queries. We refer to such data as the *query-data.* 3) Finally, the sink may send out one or more *queries* requesting information about the detection of a particular target. The sensors can then match targets detected with the prestored images and respond accordingly.

The problem of the reliable data delivery in multihop wireless networks has been addressed by several existing works in the context of wireless ad hoc networks [7]. However, such approaches do not directly apply to the environment of WSN because of three unique challenges imposed by the following considerations: 1) *Environment considerations.* The constraints imposed by a WSN environment are substantially different from those imposed by other types of multihop wireless networks. A few examples include the limited lifetime of network nodes, the scarcity of bandwidth and energy, and the size of the network itself. 2) *Message considerations.* Although most approaches for group reliable transport over multihop wireless networks in related works consider large-sized messages (spanning several packets), most messages in a sensor network might be small-sized *queries.* This raises fundamental issues on what kind of loss recovery schemes can be employed. 3) *Reliability considerations.* The notion of reliability that is traditionally prevalent is that of a simple 100 percent reliable data delivery. However, WSN might require other notions of reliability ranging from the 100 percent reliable delivery to only a subregion of the network to partial probabilistic reliability for scoped-resolution-based querying.

- *S.-J. Park is with the Louisiana State University, 289 Coates Hall, Baton Rouge, LA 70809. E-mail: sjpark@csc.lsu.edu.*
- *R. Vedantham is with the Georgia Institute of Technology, Van Leer, 777 Atlantic Dr., Atlanta, GA 30332. E-mail: ramv@ece.gatech.edu.*
- *R. Sivakumar is with the Georgia Institute of Technology, 5164 Centergy, 75 Fifth Street, Atlanta, GA 30308. E-mail: siva@ece.gatech.edu.*
- *I.F. Akyildiz is with the Georgia Institute of Technology, 5170 Centergy, 75 Fifth Street, Atlanta, GA 30308. E-mail: ian@ece.gatech.edu.*

Manuscript received 15 Nov. 2005; revised 13 Oct. 2006; accepted 21 Mar. 2007; published online 5 June 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0338-1105.
Digital Object Identifier no. 10.1109/TMC.2007.70707.

In this paper, we address the above challenges and present an approach called GARUDA (a mythological bird that reliably transported gods) that provides reliable point-to-multipoint data delivery from a sink to sensors. GARUDA is scalable with respect to network size, message characteristics, loss rate, and reliability semantics and consists of the following elements as the cornerstones of its design:

1. an efficient pulsing-based solution for reliable short-message delivery,
2. a virtual infrastructure called the *core*, which approximates an optimal assignment of local designated servers and is instantaneously constructed during the course of a *single* packet flood,
3. a two-stage negative acknowledgment (NACK)-based recovery process that minimizes the overheads of the retransmission process and performs out-of-sequence forwarding to leverage the significant spatial reuse possible in a WSN, and
4. a simple candidacy based solution to support the different notions of reliability.

We evaluate performance by showing both macroscopic and microscopic results. By providing these services with one framework, GARUDA can reduce the size of the protocol inside each sensor node.

The paper is organized as follows: Section 2 defines basic assumptions, motivates fundamental problems for the reliable data delivery, and discusses the related work. Section 3 motivates the problem of the downstream reliability, identifies the key goals, and discusses the challenges associated with realizing the goals. Section 4 describes the various design elements in the GARUDA framework, and Section 5 presents the proposed framework approach for achieving the downstream reliability to all sensors. Section 6 describes the framework for supporting the reliability variants. Section 7 compares the performance of the proposed framework with that of existing approaches. Finally, Section 8 concludes the paper.

2 Background and Related Work

We first confine the robust data delivery problem to a simple and specific reliable delivery problem with several assumptions. We then show that the inherent redundancy in sensor networks cannot guarantee any strict reliability semantics due to a variety of reasons. We argue that robustness to losses is a necessary condition in order to conserve energy since unreliable data delivery can increase energy consumption.

2.1 Assumptions

- *Downstream reliability.* Although there are many mission critical applications requiring the reliability in both the upstream and the downstream, we restrict the scope of this paper to the downstream reliability.
- *Communication and node failures.* A scheme addressing the reliability in the environment of WSN has to deal with 1) communication failures and 2) node failures. The proposed algorithm will handle both communication and node failures as we elaborate in Section 5.
- *100 percent reliable message delivery.* The reliability in WSN can have several dimensions as we mention in Section 3. At first, we focus on a basic framework that provides 100 percent reliability to all sensors. We then extend the basic framework to cover all the semantics in Section 6.
- *Message size.* We assume that the message size consists of one or more packets. It is interesting to note that, for one of the types of messages, the query, it is likely that the message size often does not exceed one packet. At the same time, support for the reliable delivery of one-packet messages poses unique challenges as we discuss in Section 3.
- *Metrics.* We consider latency, retransmission overhead, and energy consumption as the metrics of interest for comparison with other approaches. The goals of GARUDA are to minimize these metrics.
- *Network model.* We assume that both the sink and the sensors in the network remain *static.* We also assume that there is exactly one sink coordinating the sensors.

2.2 Observations

Since WSNs are characterized by a high degree of redundancy motivated by the need to extend the lifetime without redeployment, it can be conjectured that the high degree of redundancy will also provide the communication reliability. However, due to the following reasons, the redundancy cannot guarantee any reliability, thus necessitating separate reliability mechanisms.

2.2.1 Wireless Channel Errors

Wireless networks are highly influenced by random channel errors due to interference and fading effects. Fig. 1a presents the percentage of network nodes receiving a message reliably with increasing random wireless channel error rate. The message size is set to 100 packets (packet size = 1 Kbytes) and the network is a 650 m $\times$ 650 m grid with 100 nodes. It was observed that the success rate (defined as the percentage of nodes receiving a message successfully) decreases from 100 percent to about 88 percent as the random channel error rate increases from 0 to 20 percent.

2.2.2 Congestion and Contention

The downstream and upstream traffic will typically share the same channel of which capacity is limited. Hence, the downstream reliability is affected by the congestion caused by the upstream traffic. Fig. 1b illustrates the effect of background traffic on the percentage of nodes receiving a message reliably under the same network environment as in Section 2.2.1 but without channel error. It was observed that the success rate decreases from 97 percent when the aggregate background traffic (created by upstream flows from all 100 sensors to the sink using a constant bit rate (CBR) source) is 25 kilobits per second (Kbps) to 76 percent when the aggregate background traffic is increased to 400 Kbps.

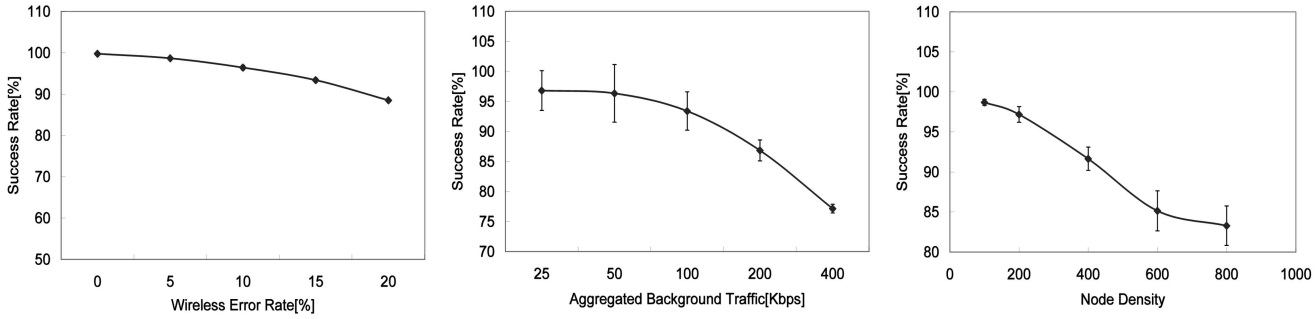

Fig. 1. Successful delivery ratio at different environments, CSMA/CA random access method (95 percent confidence interval over 20 simulation runs). (a) Wireless error rates. (b) Background traffic loads. (c) Number of nodes.

2.2.3 Broadcast Storm

Broadcast storm is the term associated with the problem that arises when flooding is performed in multihop wireless networks through a series of local broadcasts [5]. Although the problem was identified in multihop wireless networks, its impact in sensor networks is higher because of the larger density of such networks. Hence, when a message from the sink is propagated as a series of local broadcasts, the problem categorized under the broadcast storm phenomenon, namely, more collisions and higher degree of contention, results in several network nodes not receiving parts of the message. Fig. 1c presents the success rate as a function of the number of nodes within a 650 m $\times$ 650 m grid. There is no background load or random wireless channel errors. The success rate drops from about 99 percent for the 100-node scenario to 83 percent for the 800-node scenario.

2.3 Related Work

To provide the robust data delivery, researchers have proposed several approaches including: 1) physical/link layer approaches such as Forward Error Correction (FEC) [8], [9], 2) media access control (MAC) layer approaches such as reliable MAC [7], and 3) transport layer approaches such as reliable multicast [10], [11] and reliable transport protocol [12], [13].

Those presented in [10], [14], [11], and [15] are reliable multicast approaches designed for wired or multihop wireless environments assuming an address-centric routing protocol and global unique node identification. Since WSNs require a data-centric routing layer without the global identification, such approaches cannot be applied to WSNs.

FEC has been an appealing approach to prevent the feedback implosion that can happen when performing a large-scale reliable multicast [8]. However, Li and Cheriton [16] evaluate the utility of FEC for reliable multicast and compare the effectiveness of FEC with that of subcasting, which involves the multicasting of a retransmitted packet by a loss recovery server over the entire subtree rooted at the recovery server. They [16] argue that FEC provides little benefit for the efficient reliable multicast protocol like [11] that uses the subcasting. Since WSNs inherently support local subcasting because of the shared nature of the wireless channel, the gain of FEC in WSNs can be argued to be minimal.

Several works have been proposed to perform efficient flooding in multihop wireless networks [5], [17]. Williams and Camp [18] classify some of these approaches as probability-based, area-based, and neighbor-knowledge-based schemes. Although such approaches improve the successful delivery rate of messages, they still cannot guarantee any strict reliability semantics that GARUDA supports.

In [19], Gandhi et al. propose a scheme that constructs a broadcast tree and schedules transmissions with a greedy strategy to minimize the latency and the number of retransmissions involved in flooding. The approach is not targeted on large-scale networks, supports only the simplest form of reliability semantics, and does not leverage the unique characteristics of sensor environments.

Pump Slowly, Fetch Quickly (PSFQ) [13] is a transport layer protocol that addresses the issue of reliability in WSNs. The key idea of PSFQ is to distribute the data from a source node by transmitting data at a relatively slow speed but allowing nodes that experience losses to recover missing data packets from immediate neighbors aggressively. However, PSFQ does not provide any reliability for single-packet messages as it uses a pure NACK-based scheme. Also, it uses in-sequence forwarding for message delivery to accomplish the pump-slowly operation.

In [20] and [12], the authors propose reliable transport layer solutions to provide some level of reliability by controlling the reporting rate of sensors or by having multiple paths between sensors and a sink. They are concerned with upstream reliable delivery from sensors to sink.

In [6], Madden et al. have designed and implemented a query processor that incorporates acquisitional techniques called TinyDB to minimize power consumption and increase the accuracy of query results. Since [6] focuses on the energy consumption and the acquisitional issue, it cannot provide different services for the reliable data delivery that this paper concentrates on.

3 Problem Definition and Challenges

The problem addressed in this work is that of reliable sink-to-sensors downstream data delivery. We restrict the focus of the work to WSN with a sink and static sensors. The problem scope includes tackling the diverse reliability semantics required in WSN. The goal is to achieve reliability

while minimizing bandwidth usage, energy consumption, and delay.

3.1 Challenges to the Downstream Reliability of WSN

3.1.1 Environment Constraints

The bandwidth and energy constraints can be tackled by minimizing the number of *retransmission* overheads to ensure reliability. This in turn will reduce both bandwidth and energy consumption due to the overheads of the reliability process. The proneness to node failures, on the other hand, should be tackled by not relying on statically constructed mechanisms (say, a broadcast tree) that do not account for the dynamics of the network. Note that "dynamic" mechanisms that periodically refresh any constructions are not desirable as the overheads due to the reliability process have to be minimized.

Another characteristic of the target environment is the scale of the network. WSN can be expected to be of a large scale in terms of the number of nodes and, hence, the diameter of the network. This means that there is a tremendous amount of *spatial reuse* possible in the network that should be tapped for achieving the best capacity and, hence, delay. However, the specific loss recovery mechanism used may severely limit such spatial reuse as we elaborate in the next section.

3.1.2 Acknowledgment (ACK)/NACK Paradox

Although the previous challenge was with regard to the constraints imposed by the environment, this challenge stems from the constraints imposed by the typical sizes of messages used at the downstream reliability. Whereas the query-data and the control code can be expected to be a large message consisting of more than a few packets, queries pose a unique problem because of small sizes.

NACKs are well established as an effective loss advertisement mechanism in multihop wireless networks, in particular, and group communication, in general, as long as the loss probabilities are not inordinately high. However, NACKs cannot handle the unique case of all packets in a message being lost at a particular node in the network. Since the node is not aware that a message is expected, it cannot possibly advertise a NACK to request retransmissions.

If the sizes of messages are large, the probability of any packet not arriving at a node will be negligible. However, for the short message types like queries consisting of a few packets, the probability that a node does not receive any packet in a message is non-negligible and hence has to be explicitly tackled. Although an ACK-based recovery scheme would address the all-packet-lost problem, its other deficiencies (in terms of ACK implosion), however, clearly prohibit it from being used.

Finally, revisiting the issue of tapping spatial reuse, the NACK-based loss recovery scheme will inherently require *in-sequence forwarding* of data by nodes in the network to prevent a NACK implosion [13]. This will clearly limit the spatial reuse achieved in the network.

3.1.3 Reliability Semantics

Our final discussion is on constraints that are imposed by the *notion of the reliability* that typical WSN will require. Two characteristics that are innate to the WSN environment

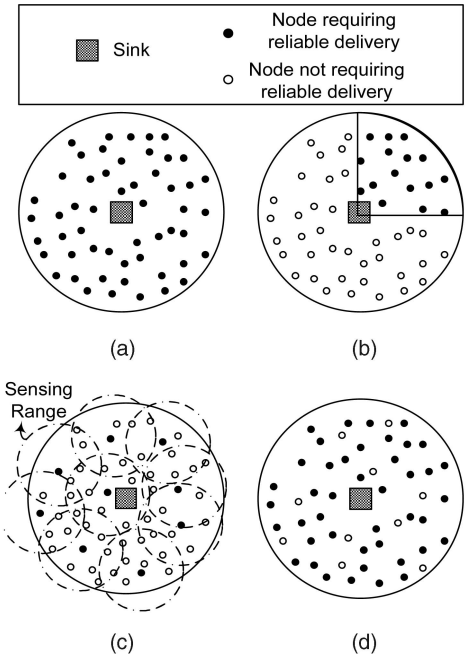

Fig. 2. Types of reliability semantics. (a) Reliable delivery to all sensors. (b) Reliable delivery to a subregion. (c) Reliable delivery to minimal sensors to cover the sensing field. (d) Probabilistic reliable delivery to 80 percent of the sensors.

are location dependency and redundancy in deployment. A query can be location dependent such as "Send temperature readings from rooms X, Y, and Z." At the same time, the redundant deployment of sensors in a field means that, in order to get reliable sensing *information*, it is not necessary for all sensors in the field to reliably deliver their locally sensed data to the sink. Furthermore, the sink might also choose to reliably deliver a message only to a probabilistic fraction of the entire network, say, as part of a sensing strategy that involves incrementally increasing resolution [3].

We thus define the reliability semantics that can be required in WSN based on the above characteristics. We classify the reliability semantics into four categories:

1. *delivery to the entire field*, which is the default semantics,
2. *delivery to sensors in a subregion of the field*, which is representative of location-based delivery,
3. *delivery to sensors such that the entire sensing field is covered*, which is representative of redundancy-aware delivery, and
4. *delivery to a probabilistic subset of sensors*, which corresponds to applications that perform resolution scoping.

Figs. 2a, 2b, 2c, and 2d illustrate categories 1 through 5, respectively. Thus, any reliability solution should support not only the default reliability semantics but also the other types of semantics that are unique to the wireless sensor environment.

4 GARUDA Design Elements

In this section, we present an overview of GARUDA's design that explicitly tackles the challenges identified in

Section 3. The centerpiece of GARUDA's design is an instantaneously constructible loss recovery infrastructure called the *core*. The *core* is an approximation of the minimum dominating set (MDS) of the network subgraph to which the reliable message delivery is desired. Although using the notion of the MDS to solve networking problems is not new [21], the contributions of this work lie in establishing the following for the specific target environment: *the relative optimality of the core for the loss recovery process*, *how the core is constructed*, *how the core is used for the loss recovery*, and *how the core is made to scalably support multiple reliable semantics*. We present a *core construction* approach that constructs the *core* during the course of a single packet flood and propose a *two-phase loss recovery* strategy that uses *out-of-sequence forwarding* and is tailored to satisfy our basic goals of minimizing the retransmission overheads and the delay. Finally, we show how a simple *candidacy*-based approach for the *core* construction can make the *core* scalably support the multiple reliability semantics.

The second cornerstone of the GARUDA design is a *pulsing*-based approach to deliver a single packet reliably to all network nodes. Recall the trade-offs identified in Section 3 for the reliable delivery of short messages. Since GARUDA can ensure the reliable delivery of the first packet of messages of any size, it is no longer vulnerable to the *all-packets-lost problem* that straightforward NACK-based schemes are susceptible to. This enables GARUDA to tap the advantages of NACK-based schemes but, at the same time, avoid any pitfalls that consequently arise.

In the rest of the section, we provide high-level overviews of the above components. For the sake of clarity, we start with discussing the details about the *core* infrastructure, assuming that the first packet is reliably delivered. Then, in Section 4.4, we present the details of reliable delivery of the first packet.

4.1 Loss Recovery Servers: Core

The *core* in GARUDA forms the set of local designated loss recovery servers that help in the loss recovery process. The challenges that hence arise are 1) how the core nodes should be chosen to minimize the retransmission overheads and 2) how the *core* can be constructed in a manner that is appropriate for the limiting characteristics (the dynamic topology change due to node failures) of the target environment.

4.1.1 Rationale of Core

To address the problem of the loss recovery server designation, we first formulate the problem as an optimization problem that has been extensively studied and investigate the applicability of the solution in the WSNs.

When a packet is broadcast from the sink to all sensors, it is not possible for all nodes to receive the packet without any errors because of a variety of reasons identified in Section 2. Further, for any two packets broadcast, the set of nodes that have not received each packet can be different. The set of nodes that have not received the packet can potentially request for retransmission from any one of the neighbors that have received it successfully. Further, the retransmission by this neighbor is sufficient to recover the loss of the same packet of all neighbors around it. If the retransmission is unsuccessful, the procedure is repeated until all losses are recovered. Thus, the problem of the optimal loss recovery server designation tries to minimize the set of retransmitting nodes ensuring that the packet is successfully received by all nodes. However, since the loss pattern for each packet can be different, the designation of optimal loss recovery servers needs to be performed for each packet.

The above optimal loss recovery server designation problem can be abstracted to one of minimum set cover (MSC) to determine the recovery server set (the set of nodes that have received the packet successfully) that covers the base set (the set of nodes that have not received the packet successfully). Further, it has been shown in [22] that the computation of the MSC is NP-complete. Thus, the optimal recovery server designation problem reduces to an MSC problem for any given loss pattern. Since the loss patterns for broadcasting two different packets can be different, the MSC has to be computed corresponding to each packet broadcast and the associated loss pattern. This implies that the set of recovery servers has to be determined for each packet broadcast.

In a real WSN environment, it is a challenge to solve the NP-complete MSC problem in a decentralized fashion. It is all the more impractical to solve the different instances of the MSC problem at the granularity of each packet broadcast. Therefore, we address the issue of loss recovery server designation with an alternate approach, which can also address the problem in a decentralized fashion.

For a given graph, a dominating set is a subset of nodes that, for every node v in a graph, either 1) v is in the dominating set or 2) a direct neighbor of v is in the dominating set. The MDS problem is to determine the dominating set with the minimum number of nodes. The MSC problem is closely related to the MDS problem and has been shown to be equivalent to the MSC problem using L-reduction. Also, the MDS problem has been shown to be NP-hard [23], [22]. Further, both of these problems find the minimum cardinality subset of nodes that cover their neighbors. Thus, the MSC problem can be thought of as a modified version of the MDS problem by applying the L-reduction transformation to the modified MDS problem.

Thus, we can address the loss recovery server designation problem by considering it as a modified MDS problem, which has the following advantage and disadvantage: 1) the advantage of the MDS problem is that the solution is agnostic to the exact loss pattern for each broadcast packet and is unique irrespective of the different instances of loss pattern for different packet transmissions and 2) the disadvantage of MDS is that the cost of determining the optimal solution for MDS is larger than that of the optimal solution of the MSC for a given loss pattern S. To make the solution practical, it is necessary to decouple the given set S from the assumptions of the problem. This necessitates a need for a practical solution of the MDS problem that will cover all the nodes irrespective of the different loss patterns.

To find the ratio between the practical approximation of the optimal solution and the optimal solution, we define two cost metrics that correspond to the number of dominating nodes in a given graph and the size of

theminimum cover set, respectively: 1) $PAPX(MDS)$ is the cost of the practical approximation of the optimal solution for the MDS problem that covers all the nodes V in the WSN and 2) $OPT(MSC)$ is the cost of the optimal solution for the MSC problem, given the knowledge of the different loss patterns. The cost is then defined by the expression $\frac{PAPX(MDS)}{OPT(MSC)}$.

Given a graph $G = (V, E)$ and a set system (X, S), we assume that the given graph G has the maximum degree G_d, which limits the maximum number of neighbors at a node. To compare the costs, we divide the problem into three cases: 1) the given set S is the subset of the dominating set $D \subseteq V$ approximated by practical solutions, 2) the given set S is the subset of the complement set $\overline{D} \subseteq V$, and 3) the other remaining case when a part of the given set S is the subset of the set D and the remaining part of the set S is the subset of the set $\overline{D}$.

Case 1. If $S \subseteq D$, each element s_i where $S = \{s_1, \cdots, s_k\}$ is located at D that guarantees the minimum number of nodes covering all nodes in the set V. The cost of $PAPX(MDS)$ to cover the set S is equal to k that is the size of the set S, although the size of D is larger than that of S. In the best case, the cost of the optimal solution of MSC is as follows:

$$OPT(MSC) \geq \frac{|S|}{G_d}, \tag{1}$$

since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of the graph G. Therefore, we can find the upper bound of the ratio as follows:

$$\frac{PAPX(MDS)}{OPT(MSC)} \leq \frac{|S|}{\frac{|S|}{G_d}} = G_d. \tag{2}$$

Case 2. If $S \subseteq \overline{D}$, each element s_i in S is a neighbor of nodes in D that guarantees the minimum number of nodes covering all nodes in V. In the worst case, the cost of $PAPX(MDS)$ to cover S is less than or equal to k, since each element s_k can be dominated by different nodes in D. Moreover, in the best case, the cost of the optimal solution of MSC is the same as in (1) since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of G. Therefore, we can find the upper bound of the ratio as in (2).

Case 3. If $S_D \subseteq D$ and $S_{\overline{D}} \subseteq \overline{D}$ such that $S = S_D \cup S_{\overline{D}}$, in the worst case, each element in S_D and $S_{\overline{D}}$ can still be dominated by a node in D. Therefore, $|S|$ number of nodes from D are required to cover S at most. In the best case, the cost of the optimal solution of MSC is the same as in (1), since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of G. Therefore, we can find the upper bound of the ratio as in (2).

For all cases, the ratio $\frac{PAPX(MDS)}{OPT(MSC)}$ is bounded by G_d, which is similar to the approximation ratio of $\frac{APX(MSC)}{OPT(MSC)} = ln(k)$, as the average value of G_d is $log(n)$ [23].

Therefore, with a reasonable approximation ratio, we can use the practical approximation of MDS to solve the MSC problem, which is the optimal solution of the loss recovery server designation problem.

4.1.2 Instantaneous Core Construction

The *core* is constructed using the first packet delivery. The reliable delivery of the first packet determines the *hop_count* of each node, which is the distance of a node from the sink. A node, which has a *hop_count* that is a multiple of three, elects itself as a core if it has not heard from any other core node. To approximate the MDS problem, we select a node at $3i$ hop distance as a core node because it can cover the other nodes at $3i + 1$ or $3i - 1$ hop distances so that it can behave like as one of the MDS in the direction from a sink to sensors.

In this fashion, the core selection procedure approximates the MDS structure in a distributed fashion. The uniqueness of the *core* lies in the following characteristics: 1) the *core* is constructed using a single-packet flood, more specifically, during the flood of the first packet, and 2) the structure of the sensor network topology is leveraged for more efficient and fair *core* construction. Such an instantaneous construction of the core nodes (during the first packet delivery of every new message) addresses any vulnerability in the network in terms of node failures occurring at the granularity of a message.

4.2 Loss Recovery Process

4.2.1 Out-of-Sequence Packet Forwarding with A-Map Propagation

In GARUDA, an out-of-sequence packet forwarding strategy is used as opposed to an in-sequence forwarding scheme. A key drawback of the in-sequence forwarding strategy is that precious downstream network resources can be left underutilized when the forwarding of higher sequence number packets is suppressed in the event of a loss. The out-of-sequence forwarding, on the other hand, can overcome this problem as nodes that have lost a packet can continue to forward any higher (or lower) sequence number packets that they might have received. However, such an approach can potentially lead to unnecessary NACK implosion, where downstream nodes will issue a chain of NACK requests for holes detected in the sequence of packets received, even when the concerned packets are not available.

To inhibit such unnecessary retransmission requests, GARUDA uses a scalable *A-map* (Availability Map) exchange between core nodes that conveys metalevel information representing the availability of packets with bits set. Any downstream core node initiates a request for a missing packet only if it receives an *A-map* from an upstream core node with the corresponding bit set. The *core* recovery phase is highly efficient as the core nodes initiate requests only when they are sure of an upstream core node having a particular packet. Although the overhead associated with the A-map is an obvious concern, the performance results in Section 7 take into account the A-map overhead and, hence, any improvements shown are after accounting for the A-map overhead.

4.2.2 Two-Stage Loss Recovery

Once the *core* is constructed, the framework employs a *two-stage recovery process* that first involves the core nodes recovering from all lost packets and, then, the recovery of lost packets at the noncore nodes. The reasons for using the two-stage recovery are threefold: 1) the number of noncore nodes will be a substantial portion of the total number of nodes in the network and, hence, precluding any contention from them is desirable; 2) when the core nodes perform retransmissions for other core nodes, holes corresponding to lost packets among the neighbors of the core nodes would also be filled with a single retransmission; and 3) when only the core nodes are performing retransmissions during the second phase, due to the nature of the *core* (ideally, no two core nodes are within two hops of each other), the chances for collisions between retransmissions from different core nodes are minimized. The two stages of recovery are described as follows:

- *Loss recovery for core nodes.* The recovery process for the core nodes is performed in parallel with the underlying default message forwarding. This is done in order to ensure that the core nodes receive all the packets in a message as quickly as possible. This parallel recovery process for the core nodes does not increase the contention in the network significantly because the fraction of core nodes is very small compared to the total number of nodes in the network and all requests and retransmissions are performed as unicast transmissions to the nearest upstream core that has a copy of the lost packet.
- *Loss recovery for noncore nodes.* The second phase of the loss recovery starts only when a noncore node overhears an *A-map* from the core node indicating that it has received all the packets in a message. Hence, the second phase does not overlap with the first phase in each local area, preventing any contention with the basic flooding mechanism and with the first phase recovery.

Although the two-phase loss recovery can potentially increase latency, we show in Section 7 that the proposed framework incurs a significantly smaller latency than competing approaches.

4.3 Multiple Reliability Semantics

In this section, we outline briefly how the *core* construction can be simply modified to account for the multiple reliability semantics identified in Section 3. We assume, without loss of generality, that a given instance of reliability semantics will require reliable delivery to a subset G_S of the nodes in the underlying graph G. Consider the subset G_S to consist of K components, where each component is connected, but the components themselves are not connected with each other. The desired infrastructure for such a setting will entail the computation of the MDS for each component and connecting the components back to the sink using a *traveling salesman path (TSP)* if bandwidth costs were the optimization criterion [24].

GARUDA uses a simple and effective technique to compute the individual MDS and connect them back to the sink using an approximation of the *shortest path tree* (SPT). Although this may incur additional bandwidth costs, note that it will have the benefit of better delay properties in addition to being implicitly constructible as we describe in Section 6. Fig. 3 shows GARUDA's solution that finds the MDS within each partition and approximates the SPT connecting all MDS to the sink.

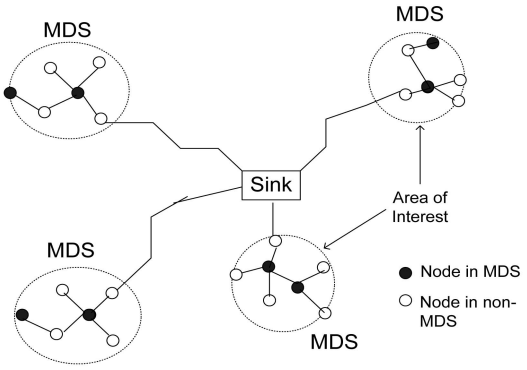

Fig. 3. Types of reliability semantics.

The MDS within each component is constructed with minor changes to the *core* construction algorithm that merely involves nodes employing a *candidacy* check before participating in the *core* construction algorithm. The candidacy check is where nodes, upon receiving the first packet, determine whether or not they belong in the subset G_S. Nodes outside G_s but required for the construction of the SPT are inducted into the core structure through a *forced candidacy* mechanism.

4.4 Reliable Single-/First-Packet Delivery

Since the NACK-based request schemes do not suffice for the single-packet delivery (or all packets are lost), we consider an ACK-based scheme as an alternative just for the first packet. However, such an approach will still incur the undesirable ACK implosion problem identified in Section 3.

GARUDA addresses the reliable delivery of the first packet using a *Wait-for-First-Packet (WFP)* pulse, which is a small finite series of short duration pulses, where the series is repeated periodically. The pulse has an amplitude that is much larger (at least 3 dB larger) than that of a regular data transmission and a period that is significantly shorter than that of a regular data transmission. By turning on a transmission radio during the minimum period when an energy detection module can detect the pulse, we can generate the single-tone pulse that is similar to the periodic pulses in [25].

Due to the characteristic of the Direct Sequence Spread Spectrum (DSSS), we can assume that the WFP pulse will not interfere with a node receiving a data packet. Therefore, the unique property of the pulse is that any node, irrespective of whether it is currently idle or receiving a regular data packet, can sense the pulse. Although we still need to change the firmware of the devices for those standards to perform energy-based detection, the notion of a pulsing mechanism has been studied and shown to be feasible [25], [26].

When a sink wants to send the first packet, the sink transmits the finite series of the WFP pulse periodically. The sensor nodes within the transmission range of the sink, upon reception of the pulses, also start pulsing with the same periodicity between two series of pulses, and this process is repeated until all the nodes start pulsing in anticipation of the reception of the first packet. The sink, after pulsing for a finite duration (to ensure that the pulses have propagated across multiple hops in the network), transmits the first packet as a regular data packet transmission and stops sending any further WFP pulses. Every sensor upon reception of the first packet also performs the same set of actions.

Essentially, the WFP signal serves two purposes: 1) it allows the sink to inform the sensors about an impending message that has reliability requirements, and 2) it enables sensors to request for retransmissions when they do not receive the first packet successfully. It might appear that resource-constrained sensors can be overloaded in terms of energy consumption and cost with the addition of the pulsing mechanism. However, we argue that the addition of the WFP signal alleviates several problems associated with the reliable message delivery and can in fact provide benefits that far outweigh the costs:

1. Since the WFP pulse is used to indicate the arrival of an impending new transmission, it does not require the same modulation scheme for the data transmissions. Additionally, the WFP pulse is robust to fading effects and interference since the pattern of periodic pulses can be differentiated from them.
2. The message advertisement scheme using WFP pulses is inherently robust to collisions, as the collisions of WFP pulse with other such pulses or data transmissions does not prevent sensors hearing the WFP pulses from inferring the impending message transmission (they still will sense that the WFP pulses are being sent) [26].
3. Unlike in the ACK-based scheme, where the ACK implosion can adversely impact the data transmissions as they do not scale well to the increasing number of nodes, the WFP pulse serves as an *implicit NACK* and (because of their small width) interferes to a very minimal extent with the regular data transmissions.
4. The energy consumption of the WFP pulse is significantly smaller than that of a regular data transmission, thus rendering any additional energy consumption to be far less than the actual energy savings because of the other benefits.

We will profile the energy savings through the use of the WFP pulses in Section 7.

5 GARUDA Framework

The details of the framework are presented with an assumption of a simple underlying flooding mechanism. However, GARUDA can as well be integrated with the flooding scheme itself. We assume that every incoming flooded packet is passed to GARUDA if it is part of a message requiring the reliability. The different components of GARUDA are explained in the chronological order that they occur when a reliable message is flooded. Hence, we first describe the details of GARUDA's pulse-based single-packet delivery mechanism and then describe the *core* construction and the loss recovery procedures. Note that the reliable single-packet delivery is leveraged for the instantaneous *core* construction.

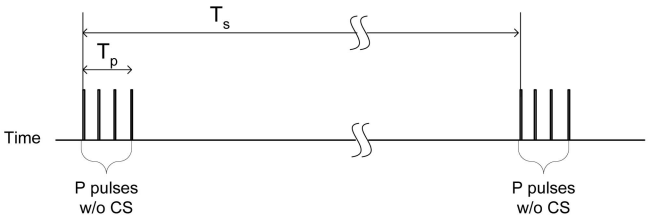

Fig. 4. Transmission time of a WFP pulse.

5.1 Single-/First-Packet Delivery

5.1.1 WFP Pulse Transmission

Since a WFP pulse can be regarded as a short-period signal that does not include any information, the transmission period of the WFP pulse is significantly smaller when compared to the transmission time T_D required for a regular data packet. Also, twice the regular transmission power is used to transmit the pulses to achieve a relative amplitude of 3 dB at the receiver (with respect to a default reception). The detection of a WFP pulse at a receiver is done based on a simple energy detection strategy that monitors changes in the amplitude of the energy of the incoming signal and the duration of any such changes [25], [26]. Note that the changes in energy can be detected even at receivers whose local channel is busy with an ongoing data reception. The only nodes that cannot hear the WFP pulses are those that are not listening (either in transmit mode or in a power-down mode).

Due to the noisy environment of WSN, pulse detection can be influenced by the wireless fading effects and interference. Therefore, the rate of false positive detections may be increased. To increase the robustness of the pulse detection, every set of pulse transmission includes p pulses transmitted consecutively within a period T_P ($T_P \ll T_D$). Fig. 4 shows the transmission scheme for the WFP pulse. Hence, receivers infer an incoming WFP signal only after detecting p pulses. The basic (and the only required) mechanism for WFP pulsing in GARUDA does not use any carrier sensing and, hence, is referred to as *forced WFP pulsing*. This ensures that nodes that need to transmit the WFP (either as an advertisement or a NACK for the first packet) can do so without having to suffer from any MAC layer starvation problems. However, such transmissions clearly increase the chances for collisions with regular data packet transmissions and, hence, are performed with a period T_s, where $T_s \gg T_D$. However, the forced pulsing in GARUDA is complimented with a carrier-sensing-based WFP and a data-packet-piggybacking-based advertisement scheme that reduce the impact of the forced WFP.

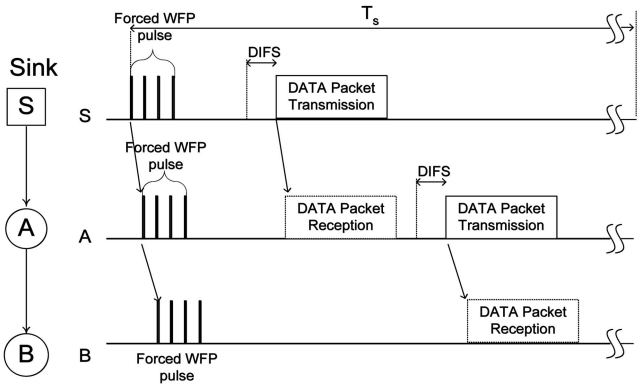

Fig. 5. Example for single-/first-packet delivery.

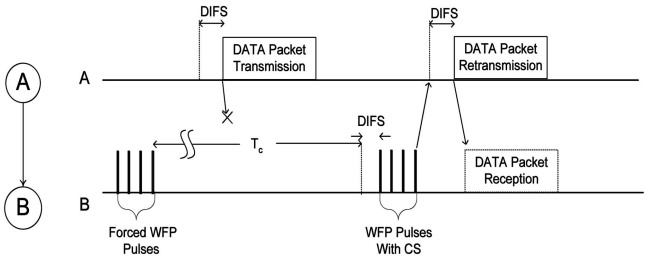

Fig. 6. Example for loss detection and recovery.

5.1.2 First-Packet Delivery

The delivery procedure for the single/first packet consists of three modes: 1) the advertisement that notifies the ensuing single/first packet to all nodes with the forced WFP pulses; 2) the delivery that sends the single/first packet through simple forwarding; and 3) the recovery that sends NACKs using WFP pulses to request for the retransmission of the single/first packet.

Fig. 5 shows the basic procedure of the single-packet or first-packet delivery. When a sink wants to initiate a reliable single-/first-packet delivery, it sends a set of forced WFP pulses without sensing the wireless channel. When neighboring sensors hear the WFP pulses, they send a set of the forced WFP pulses immediately. After a deterministic period set based on the diameter of the network, the sink transmits the single/first data packet subject to the medium access scheme (for example, Carrier Sense Multiple Access (CSMA)). If the node receives the single/first packet, it changes its operation from the advertisement mode to the delivery mode by halting the WFP pulses and by sending the single/first data packet after carrier sensing.

However, if the single/first packet is lost, nodes will continue to transmit the WFP pulses, which in turn trigger retransmissions shown in Fig. 6. Since the forced WFP pulses sent every T_s period play the role of a NACK signal, node B will wait for a duration of at least T_s to send the next set of forced WFP pulses. Therefore, the latency for the single-/first-packet delivery is directly dependent upon T_s.

To reduce the latency, GARUDA uses another kind of WFP pulse that a node sends after a regular carrier-sensing operation. Node B sends p number of WFP pulses after carrier-sensing (WFP_{cs}) opportunistically (unless it has received the single/first packet) with a period T_c that is smaller than T_s. The period T_c[1] should be proportional to the hop distance of the node B from the sink because a node should wait until the upstream nodes receive a single or the first packet.

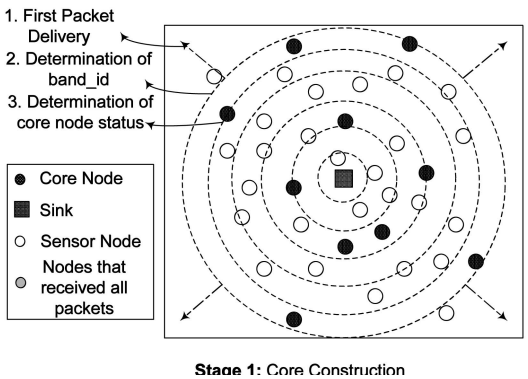

Fig. 7. Instantaneous core construction in GARUDA.

Since a node senses the state of channel before transmitting WFP_{cs} pulses, the WFP_{cs} pulses have a lesser probability of colliding with data packets than the WFP pulses. When a node gets to transmit WFP_{cs} pulses, it resets the timer corresponding to the T_s time period for the forced WFP pulses.

After a node hears a series of the WFP pulses, it will wait for a DIFS period weighted against the number of the WFP pulses. This gives more chances for a node to have a higher number of neighboring nodes that send the WFP pulses to retransmit the first packet.

An opportunistic optimization at GARUDA is the piggybacking of the NACK information inside the regular packet. The NACK is merely the sequence number of the last message ID the node has received thus far. Any neighbor that is aware of a greater message ID and has the corresponding first packet then retransmits. We refer to this as an implicit NACK mechanism.

5.2 Instantaneous Core Construction

Assuming a network topology shown in Fig. 7, the first-packet delivery establishes *band-IDs* for nodes based on the hop distance that they perceive from the sink. We consider all nodes with the same band-ID as forming a "band" with a certain ID. The bands can be viewed as concentric circles around the sink. In addition, every core node in the $3(i+1)$ band knows of at least one core node in the $3i$ band.

5.2.1 Core Construction Procedure

Sink:

- When the sink sends the first packet, it stamps the packet with a "band-ID" (bId) of 0. When a sensor receives the first packet, it increments its bId by one and sets the resulting value as its own band-ID. The band-ID is representative of the approximate number of hops from the sink to the sensor.

1. T_C is heuristically set to $i \times \Delta \times T_D$, where i is the hop distance from a sink to a node, and Δ is the maximum node degree.

Nodes in $3i$ bands:

- Only sensors located at a $3i$ band, where i is a positive integer, are allowed to elect themselves as core nodes.
- When a sensor S_0 with a band-ID of the form $3i$ forwards the packet (after a random waiting delay from the time it received the packet), it chooses itself as a core node if it had not heard from any other core node in the same band. Once a node chooses itself as a core node, all packet transmissions (including the first) carry information indicating the same.
- If any node in the core band that has not selected itself to be a core receives a core solicitation message explicitly, it chooses itself as a core node at that stage.
- Every core node S_3 in the $3(i+1)$ band should also know of at least one core in the $3i$ band. If it receives the first packet through a core in the $3i$ band, it can determine this information implicitly as every packet carries the previously visited core node's identifier bId and *A-map*. However, to tackle a condition where this does not happen, S_3 maintains information about the node (S_2) it received the first packet from, and the S_2 node maintains information from the node (S_1) it received the first packet from. After a duration equal to the core election timer, S_3 sends an explicit *upstream core solicitation* message to S_2, which in turn forwards the message to S_1. Note that, by this time, S_1 already has chosen a core node, and it responds with the relevant information.

Nodes in $3i+1$ bands:

- When a sensor S_1 with a band-ID of the form $3i+1$ receives the first packet, it checks to see if the packet arrived from a core node or from a noncore node. If the source S_0 was a core node, S_1 sets its core node as S_0. Otherwise, it sets S_0 as a candidate core node and starts a core election timer that is set to a value larger than the retransmission timer for the first-packet delivery. If S_1 hears from a core node S_0' before the core election timer expires, it sets its core node to S_0'. However, if the core election timer expires before hearing from any other core node, it sets S_0 as its core node and sends a unicast message to S_0 informing it of the decision.

Nodes in $3i+2$ bands:

- When a sensor S_2 with a band-ID of the form $3i+2$ receives the first packet, it cannot (at that point) know of any $3(i+1)$ sensor. Hence, it forwards the packet without choosing its core node but starts its core election timer. If it hears from a core node in the $3(i+1)$ band before the timer expires, it chooses the node as its core node. Otherwise, it arbitrarily picks any of the sensors that it heard from in the $3(i+1)$ band as its core node and informs the node of its decision through a unicast message. If it so happens that S_2 does not hear from any node in the $3(i+1)$ band, it sends an anycast *core solicitation* message with only the target band-ID set to $3(i+1)$. Any node in the $3(i+1)$ band that receives the anycast message is allowed to respond after a random waiting delay. The delay is set to a smaller value for core nodes to facilitate the reuse of an already elected core node.
- A boundary condition that arises when a sensor with a band-ID of $3i+2$ is right at the edge of the network is handled by making the band act just as a candidate core band $(3i)$. Such a condition can be detected when nodes in that band do not receive any response for the anycast core solicitation.

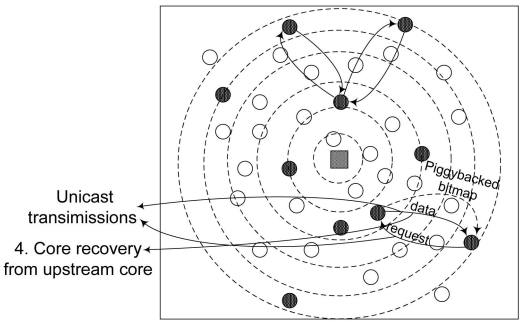

Fig. 8. Loss recovery for core nodes in GARUDA.

5.3 Two-Phase Loss Recovery

5.3.1 Loss Recovery for Core Nodes

a. *Loss detection.* When a core node receives an out-of-sequence packet, the core node infers a loss, and it sends a request to an upstream core node only if it is notified through an *A-map* that the missing packet is available at the upstream core node.

b. *Loss recovery.* When a core node receives a unicast request from a downstream core node, it performs a unicast retransmission for the request. Fig. 8 shows the loss detection and the loss recovery between core nodes at the $3i$ band and core nodes at the $3(i+1)$ band. If any of the noncore nodes on the path of the unicast request has the requested packet, it intercepts the request and retransmits the requested packet.

The use of the *A-map* is central to the core recovery process. For the sake of brevity, we assume that the *A-map* is capable of representing all packets of a message irrespective of the message size. The core recovery process works as follows:

Upstream core nodes:

- A core node, when it forwards a packet, stamps on the packet the following metainformation: $(C_{id}, A-map, bId, vFlag)$, which consists of the core node's identifier, bit map, band-ID, and valid flag, respectively. The valid flag is used by a recipient core node to determine whether the path in the metainformation is valid or not.
- When a core node receives a retransmission request, it responds with unicast retransmissions of the available packets. The unicast retransmission is

implemented with multiple forwarding at intermediate noncore nodes that know upstream and downstream core nodes.

Intermediate noncore nodes:

- A noncore node NC_{id} that forwards a packet leaves the *A-map* information untouched but adds its identifier as follows: $(C_{id} + NC_{id}, A - map, bId)$. If the number of the identifiers in the incoming packet is equal to three, the noncore node does not add its identifier and sets the $vFlag$ to NULL.

Downstream core nodes:

- Thus, when a core node receives the metainformation, it knows of not only what packets the source core node has, but also the path it can use to request for a retransmission. If the $vFlag$ is NULL, the core node still uses the *A-map* information but falls back on any earlier cached path to the relevant core node for issuing the request.
- Each core node maintains two *A-map*s locally: $myBM$, representing the successfully received packets, and $totBM$, representing both the received and the requested packets.
- When a core node receives an incoming *A-map* $(inBM)$, it checks if the *A-map* is from a valid source. If the source is valid, it checks if the A-map conveys the availability of a packet that has neither been received nor been requested. If at least one such packet is available, the node creates a request *A-map*, updates its $totBM$, and sends the request. It also starts an expiry timer for the request.
- For a successful packet reception, the core node updates its $totBM$ and $myBM$. Also, when a timer expiry occurs for a request, $totBM$ is updated accordingly.
- When a core node does not hear an *A-map* from any of its upstream core nodes for a specified duration (*core presence timer* set to a value larger than three-hop round-trip time), it issues a request to the default upstream core node to which the upstream core node responds with its current *A-map*.

5.3.2 Loss Recovery for Noncore Nodes

A noncore node snoops all (re)transmissions from its core node. Once it observes an *A-map* from its core node with all the bits set, it enters the noncore recovery phase by initiating retransmission requests to the core node. Alternatively, if it does not hear from its core node for the period *core presence timer*, it sends an explicit request to the core node to which the core node responds with its current *A-map* Fig. 9 presents the loss detection and recovery between noncore nodes and a core node. Since all retransmissions from the core nodes are snooped by the noncore nodes, redundant retransmissions for the same loss are removed.

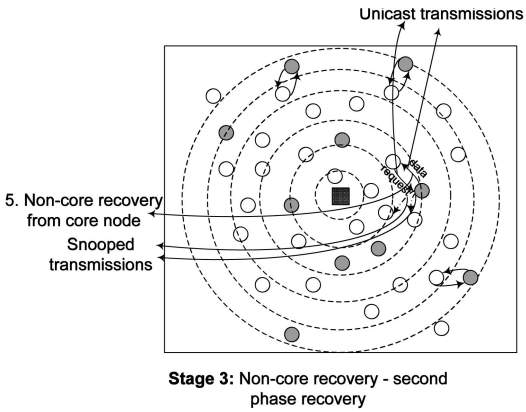

Fig. 9. Loss recovery for noncore nodes in GARUDA.

6 Supporting Other Reliability Semantics

In this section, we revisit the GARUDA design and show how it can accommodate the other reliability semantics. Specifically, we discuss three variants in terms of the reliability semantics: 1) reliable delivery to all nodes within a subregion, 2) reliable delivery to the minimal number of sensors required to cover entire sensing area, and 3) reliable delivery to p percent of the nodes.

The fundamental difference between the context in Section 5 and in the above variants is that only a subset of the nodes in the network requires reliable delivery. The variants differ in *which subset of nodes* receive the message delivery. We refer to the problem of determining the subset as the *candidacy* problem. Also, in all of the solutions discussed, the first packet is always delivered to all nodes in the network. All subsequent packets are delivered based on the candidacy. Generically, the solutions to the three variants use three common elements to tackle the other reliability semantics:

- The first packet carries information to identify the eligibility for candidate nodes that should receive the entire message reliably. For example, in the case of reliability within a subregion, the first packet may carry a coordinate-based description of the subregion.
- Participation in the *core* construction is limited to only those nodes that have chosen themselves as candidates. The other aspects of the *core* construction still remain the same (nodes only in the $3i$ bands can select themselves as core nodes). At the end of the *core* construction, each component of the candidate subgraph G_S has its own core.
- The last element is that of *forced candidacy* to enable the *core* of the different components to be connected back to the sink. Thus, noncandidate nodes in the $3i$ bands on the path from each component to the sink are forced to participate as candidate core nodes to ensure connectivity. The forced candidacy is actually achievable in GARUDA with very minimal changes to its original design. Essentially, noncandidate nodes in core bands, if they would have otherwise chosen themselves as core nodes, identify themselves as noncandidate core nodes when the first packet is forwarded. A downstream candidate core node that has not heard from any other candidate upstream core node explicitly requests the upstream noncandidate core node to become a candidate.

Through this process, a structure that is an approximation of independent MDSs (within each component of G_S) connected through an SPT is achieved.

6.1 Reliable Delivery within a Subregion

As we motivate in Section 3, it is likely that the sink requires reliable delivery of a query or a message only to sensors within a specific subregion of the network area. We assume that the subregion can be specified in the form of coordinates. Without loss of generality, we assume that the subregion is rectangular in shape (although the GARUDA design does not have any such limitations). The subregion can either be contiguous or noncontiguous with the region occupied by the sink.

The desired subregion coordinates are piggybacked on the first packet sent by the sink. Each sensor receiving the first packet can thus determine locally whether it is a candidate or not, based on its own location and the desired subregion. Once the candidacy is determined, the behavior of sensors is the same as that described in Section 5, except if the sensor was to be on a core band. Whereas in the default operation, a sensor does not choose itself as a core node only if it hears from another core node before it transmits; under this variant, a sensor does not choose itself as a core node if it is not a candidate irrespective of the other conditions. Note that this does not mean that such a sensor can later be forced to become a core node, as we elaborate next.

6.2 Reliable Delivery to Cover the Sensing Field

This variant requires reliable delivery while remaining aware of the inherent redundancy in the sensor network deployment. Specifically, under this variant, reliable delivery needs to be performed only to a minimal subset of the sensors such that the entire sensing field is covered. For purposes of this discussion, we assume that the sensing range S is always less than or equal to the transmission range R.

Unlike in the previous variant where the candidacy of each node is determined locally, in this variant, coordination between nodes is required to eliminate sensors, which are covering a region already covered by other sensors, from the candidacy. In GARUDA, the core nodes are best equipped to perform such coordination as they are immediately adjacent to all noncore nodes and, under ideal conditions, are at least a distance of $2R$ away from the nearest core node (which gives a core node a virtual "ownership" of at least the sensing region defined by its transmission range). Thus, noncore nodes seek permission from their respective core nodes to become candidates. Each core node keeps track of the coverage of the region defined by the square of side $2(S+T)$. It provides permission to a seeking noncore node only when the node can cover an area not already covered inside the square. Note that, given our assumptions about S and T, no noncore node within a core node's transmission range can have a sensing coverage area that even lies partially outside the square.

All core nodes implicitly become candidates. This is reasonable even without any coordination with other nearby core nodes as, under ideal conditions, the distance between a core node and its nearby core nodes will be $2R$, which in turn means that a core node can choose itself as a candidate without concern of overlapping with a nearby core node's sensing region.

6.3 Reliable Delivery to a Probabilistic Subset

This variant supports the reliable message delivery to p percent of the network sensors. Such semantics is useful when the sink intends to perform *scoped* sensing. For instance, the sink can at the outset decide to sense only 25 percent of the field, with the intent of increasing the sensed region only upon some triggers detected during the preliminary sensing. Just as in the case of a delivery within a subregion, determining the candidacy in this variant is purely a local process. When a sensor receives the first packet, it chooses itself as a candidate with a probability of p. If the sensor is on a core band and decides not to be a candidate, it does not choose itself as a core node irrespective of the other conditions.

7 Performance Evaluation

7.1 Simulation Environment

For all *ns2*-based experiments:

1. The first 100 nodes are placed in a grid fashion within a 650 m $\times$ 650 m square area to ensure connectivity, whereas the remaining nodes are randomly deployed within that area, and the sink node is located at the center of one of the edges of the square.
2. The transmission range of each node is 67 m [27].
3. The channel capacity is 1 Mbps.
4. Each message consists of 100 packets transmitted at the rate of 25 packets per second (except for the single-packet-delivery part), and the size of packet is 1 Kbyte.

(Although the above environment seems be different to sensor networks, we argue that the relative performance difference among GARUDA and the others, such as in-sequence and out-of-sequence delivery schemes, still will be maintained at the low-speed physical layer.) CSMA/Collision Avoidance (CA) is used as the MAC protocol. We use basic flooding as the routing protocol. All the simulation results are shown after averaging the metrics over 20 randomly generated topologies and calculating 95 percent confidence intervals. As described in Section 3, losses can occur due to wireless channel errors or collisions among transmissions. To emulate the two types of losses, we choose a fixed packet loss rate of 5 percent for wireless channel error and vary the number of nodes in the network (and, hence, the network density), which in turn increases the degree of contention in the network.

7.2 Evaluation of Single-Packet Delivery

7.2.1 Latency

The latency involved in receiving a single packet reliably with the increasing number of sensors is presented in Fig. 10a for both GARUDA and the ACK-based scheme. The latency of the proposed scheme is significantly smaller because of the WFP pulse, which is essentially an implicit NACK, thus not increasing the load in the network. We also

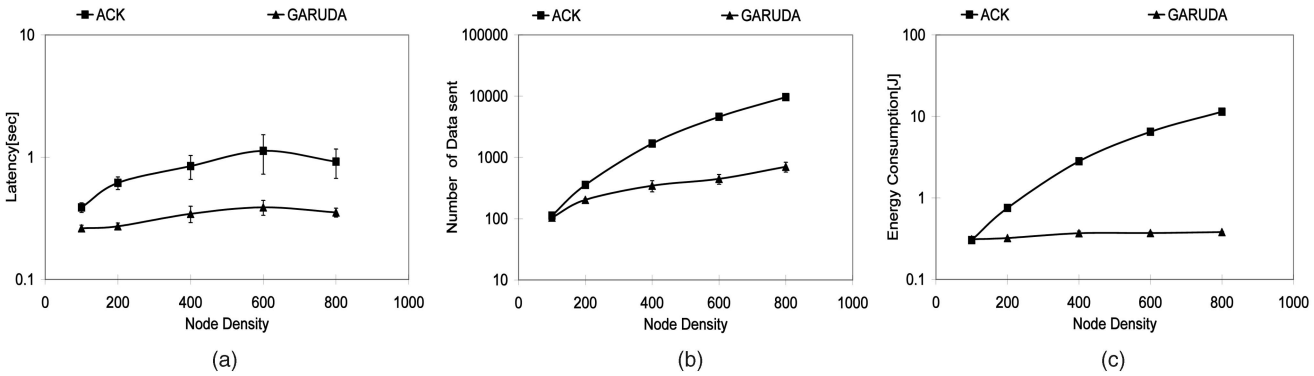

Fig. 10. Performance evaluation of GARUDA: Single-packet delivery. (a) Latency. (b) Number of data packet sent. (c) Energy consumption per node.

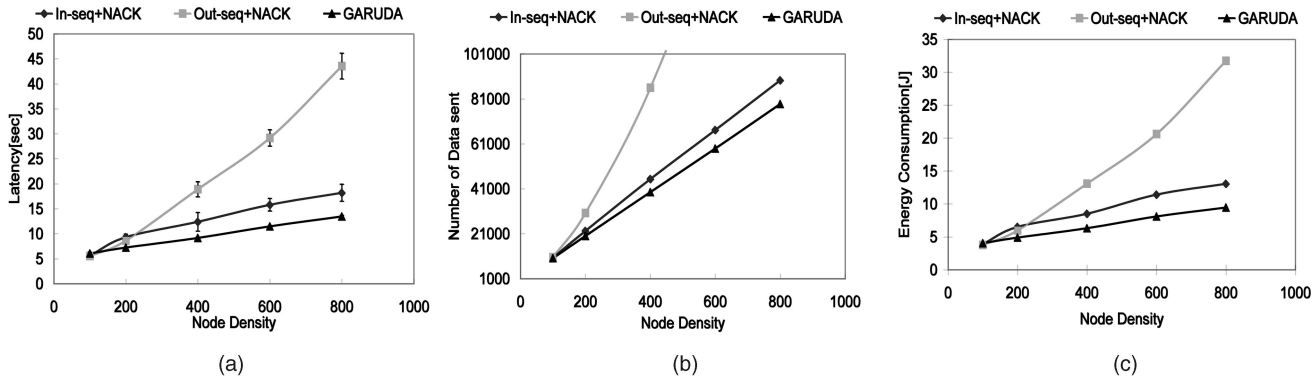

Fig. 11. Performance evaluation of GARUDA: multiple-packet delivery. (a) Latency. (b) Number of data packets sent. (c) Energy consumption per node.

see that the latency scales well with the increase in the number of nodes because of the same reason. However, in the ACK-based scheme, the latency is higher because there is an explicit ACK feedback to the sender thus increasing the traffic and, thereby, the collisions.

7.2.2 Number of Data Sent

Fig. 10b shows the number of data sent by GARUDA and the ACK-based scheme. It is interesting to note that, in the proposed framework, the number of data sent increases more or less linearly (with a slope of 1 approximately) as the number of nodes is increased. The reasons can be attributed to the implicit NACK scheme, which alleviates congestion-related losses, and the inherent redundancy, and the broadcast nature of the flooding process ensures that the packet is received successfully without any need for retransmission even in the presence of losses. For the ACK-based scheme, the number of data packets sent is appreciably higher and shows a nonlinear increasing trend with increasing number of nodes in the network. This is again because of the increased load in the network due to the presence of ACK transmissions, thus increasing the losses.

7.2.3 Energy Consumption

The energy consumptions for each scheme are shown in Fig. 10c. The energy consumption of GARUDA is significantly smaller than the ACK-based scheme even though it uses a WFP pulse. This is because of two reasons: First, the duration of the WFP pulse is insignificant compared to that of data packet transmissions. In fact, the duration of these WFP pulses can be as low as 15-20 μs in order to recognize them [26]. Second, the WFP pulses themselves do not suffer from any implosion while they address the ACK implosion problem. In fact, the energy consumed shows a linear increase with the increasing number of nodes. However, the ACK-based scheme suffers from the NACK implosion problem because energy consumption per node increases with increasing node density.

7.3 Evaluation of Multiple-Packet Delivery

To compare the performance of GARUDA for multiple-packet delivery, we have implemented two simple reliable transport protocols that allow in-sequence and out-of-sequence forwarding, respectively, coupled with NACK-based error detection and nondesignated local recovery servers.

7.3.1 Latency

Fig. 11a shows the latency for 100 percent delivery as a function of increasing the number of nodes. The proposed framework has significantly lower latencies compared to the other two schemes when the node density is increased. The reasons for reduced latencies are twofold: the advantage gained by having a local designated server as opposed to a nondesignated one, which reduces the amount of data sent, and the advantage gained by using out-of-sequence forwarding but without the NACK implosion problem, which increases the spatial reuse in the network. The

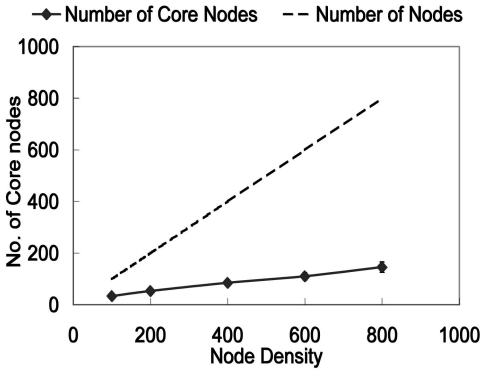

Fig. 12. Number of core nodes versus total number of nodes.

latency of the out of sequence with the NACK scheme is significantly higher at higher node densities and increases at a much faster rate than the other two schemes because of the NACK implosion problem. Although our *core* construction scheme uses an out-of-sequence delivery, we piggyback the *A-map* of the core node along with the transmission of each packet, which allows the other dependent nodes to wait for the core to recover from all losses prior to any retransmission requests, thus eliminating the NACK implosion problem.

7.3.2 Number of Data Sent

The numbers of data sent for all three schemes are presented in Fig. 11b. Among the three schemes, GARUDA performs the best, followed by the in sequence with NACK and the out of sequence with NACK schemes. The number of packets sent in GARUDA is about 10 percent lower than that of the in sequence with NACK scheme for node densities of 400, 600, and 800 and 55 percent to 80 percent lower when compared with the out of sequence with NACK scheme. The reason for the significantly better performance of GARUDA is again mainly due to the improvement gained by having a designated recovery server as opposed to a nondesignated server and the *A-map* structure propagation.

7.3.3 Energy Consumption per Node

The average energy consumed per node is significantly smaller for GARUDA when compared to the other two cases (Fig. 11c). The average energy consumed for all three cases is proportional to the number of transmissions, which is the sum of the number of requests sent and the number of data sent per node. Since the sum of the number of requests and data sent is the least for GARUDA, the energy consumed per node is also significantly less. In fact, results indicate that the energy consumed per node is about 30 percent less compared to the in-sequence case and about 80 percent less compared to the out-of-sequence scheme for the 800-node scenario.

7.4 Microscopic Analysis

7.4.1 Optimality of the Core

Since the core nodes approximate an MDS, an obvious question is how the *core* construction is set up in a way to minimize the number of core nodes. Ideally, for any given core node, there should not be any other core node in its 2-hop neighborhood. The proposed framework attempts to achieve this condition by using a two-pronged approach: 1) only nodes in $3i$ bands (core bands) are allowed to contend to become a core node, and 2) of the nodes that belong to the core bands, only nodes that have not heard from any other core node from its band are allowed to choose themselves as core nodes. Fig. 12 shows the number of core nodes as the node density is increased from 100 to 800. As we can see, the number of core nodes decreases from 30 percent when the node density is 100 to about 13 percent when the node density is 800.

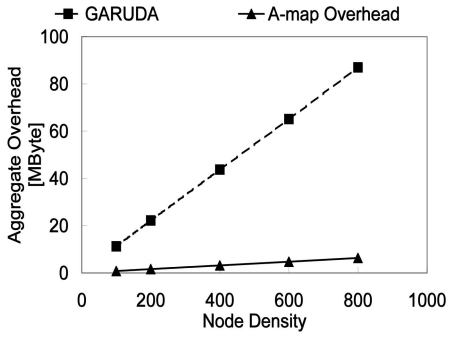

Fig. 13. Microscopic analysis: the *A-map* overhead.

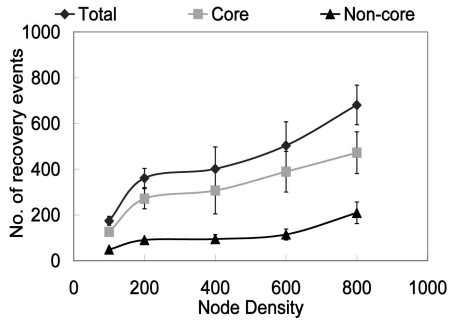

Fig. 14. Microscopic analysis: the number of recovery events.

7.4.2 A-Map Overhead

The second important aspect in the GARUDA framework is the overhead incurred by *A-map* transmission by the core nodes while sending both data and requests and the noncore nodes while sending the requests only. Although we do not expect the *A-map* overhead to be a problem for noncore nodes as their recovery happens only after their corresponding core nodes have recovered from all loses, it is an issue for the core nodes. However, Fig. 13 indicates otherwise when the *A-map* overhead is compared with the total data sent by the GARUDA scheme. There are two main reasons for this: First, the number of core nodes was only a small fraction of the total number of nodes (10-30 percent), and second, the number of requests was substantially lower (less than 1 percent) than the amount of data. In fact, in Fig. 13, we see that the *A-map* overhead was only 0-3 percent of the total amount of data sent.

7.4.3 Number of Recovery Events

We investigate the number of recovery events for core and noncore nodes and compare it with the total number of recovery events (shown in Fig. 14). This result helps us

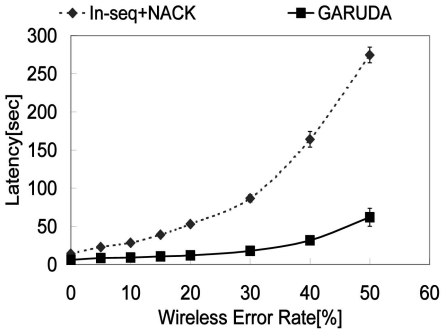

Fig. 15. Latency of GARUDA for different loss rates.

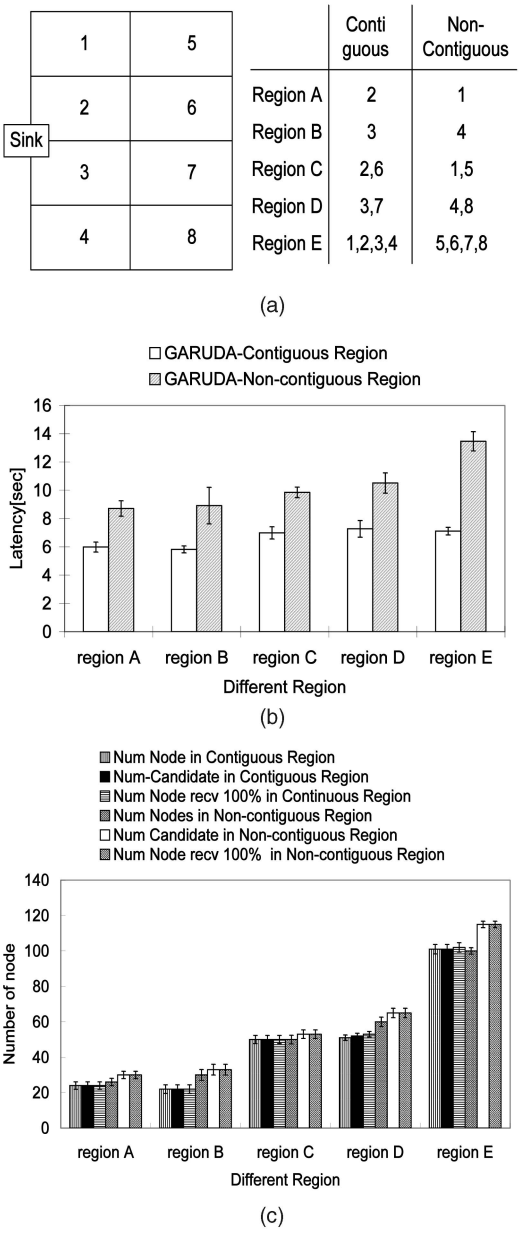

Fig. 16. Reliable delivery to all sensors in a subregion. (a) Layout of subregions. (b) Latency for different subregions. (c) Number of nodes requiring eliable delivery.

understand the two-phase recovery process better. It shows that the core recovery process (first-phase recovery) was

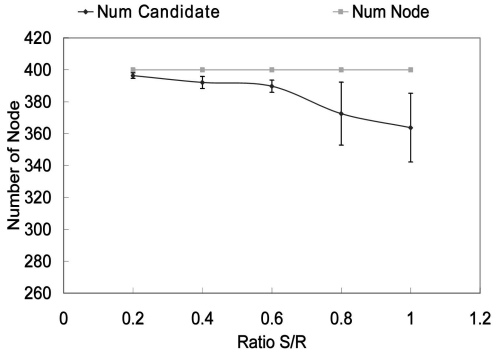

Fig. 17. Reliable delivery to the minimal number of sensors in a region.

two times more likely than the noncore recovery process (second phase) since noncore nodes were allowed to snoop recovery packets during the first recovery phase.

7.4.4 Effect of Random Wireless Errors

We have compared GARUDA with the in-sequence scheme with NACK for packet error rates ranging from 0 percent to 50 percent. For a fair comparison between the results of GARUDA and those presented in [13], a linear topology consisting of 21 sensors was used in the simulation. Fig. 15 shows that the latency of the GARUDA was much shorter than that of the in-sequence scheme with NACK, and the difference between them became larger with the increasing error rate. Although we assume a severe environment (up to 50 percent of error rate), the latency of the GARUDA shown is almost constant. GARUDA, therefore, is more adequate to WSNs since WSNs experience a higher error rate than other wireless networks.

7.5 Evaluation of Variants

7.5.1 Reliable Delivery within a Subregion

Figs. 16a, 16b, and 16c present the performance results for the first variant for a 200-node 650 m $\times$ 650 m network with a transmission range of 67 m per node. Fig. 16a shows the partitioning of the network grid into subregions. Fig. 16b shows the latency incurred with increasing number of regions for both contiguous and noncontiguous regions, respectively. Although it is obvious that the latency increases with increasing number of regions, an interesting observation is that the latency for the non-contiguous regions scenario is always more. Recall that this is due to the latency involved in noncandidates being forced into candidacy. Fig. 16c shows the number of data packets transmitted for the same scenarios. For the contiguous regions scenario, the achieved number of candidates is very close to the ideal number of candidates. However, for the noncontiguous regions, the achieved numbers are higher due to the forced candidacy of nodes to achieve connectivity.

7.5.2 Reliable Delivery to the Minimal Set of Sensors

Fig. 17 shows the number of nodes selected as candidates for the second variant. It can be observed that the number of nodes chosen decreases with the increasing ratio $\frac{S}{R}$. The decrease is not much for the smaller values of $\frac{S}{R}$ because, for the scenario considered (400 nodes in a 650 m $\times$ 650 m grid

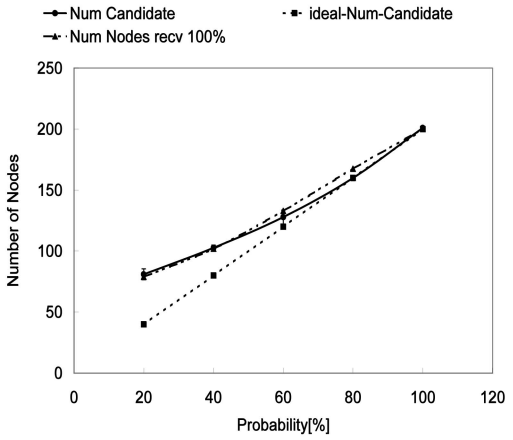

Fig. 18. Probabilistic reliable delivery.

with a transmission range of 67 m), the minimum value for $\frac{S}{R}$ required to cover the entire area is approximately 0.5. As the ratio of $\frac{S}{R}$ increases beyond 0.6, we see a more pronounced decrease in the number of candidate nodes. This is because the overlap area becomes more pronounced as the sensing range approaches the transmission range.

7.5.3 Reliable Delivery to a Probabilistic Subset

Fig. 18 presents simulation results for the third variant. The scenario considered is that of 200 nodes in a 650 m $\times$ 650 m grid, with nodes having a transmission range of 67 m. The number of candidate nodes chosen with increasing probability is shown. It can be seen that, at lower probabilities, the achieved number of candidates is larger than that of the expected number due to the forced candidacy of nodes to achieve connectivity. However, for larger probabilities (≥ 50) percent, the achieved number of candidate nodes closely approximates the ideal values.

8 Conclusions

We have proposed a new framework to provide the sink-to-sensors reliability in WSNs. We have identified several challenges to provide sink-to-sensors reliability and addressed the challenges by proposing key elements:

1. a WFP pulse,
2. a *core* structure approximating the MDS,
3. an instantaneously constructible optimal *core* structure,
4. an availability bitmap, and
5. a two-stage recovery process.

Note that, although we have proposed an effective way to realize the WFP pulse in band, it is equally possible to use an out-of-band signaling in scenarios where a pilot radio is available. We have also identified three new types of reliability semantics unique to a downstream sensor environment and elaborated how our proposed framework can provide reliability to such variants. We have shown through ns2-based simulations that the proposed framework performs significantly better than the basic schemes proposed thus far in terms of latency and energy consumption. Our future directions of work include extending the proposed framework to environments with mobility and in the presence of multiple sinks.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks J.*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. ACM MobiCom*, pp. 174-185, Aug. 1999.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom*, pp. 56-67, Aug. 2000.

[4] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," *Proc. ACM MobiCom*, pp. 271-278, Aug. 1999.

[5] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. ACM MobiCom*, pp. 151-162, Aug. 1999.

[6] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "TinyDB: An Acqusitional Query Processing System for Sensor Networks," *ACM Trans. Database Systems*, vol. 30, no. 1, pp. 122-173, Mar. 2005.

[7] K. Tang and M. Gerla, "Mac Reliable Broadcast in Ad Hoc Networks," *Proc. Conf. Military Comm. (MILCOM '01)*, pp. 1008-1013, Aug. 2001.

[8] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *Proc. ACM Conf. Applications, Technologies, Architectures and Protocols for Computer Comm. (SIGCOMM '98)*, pp. 56-67, Oct. 1998.

[9] S. Lin and D.J. Costello, *Error Control Coding: Fundamentals and Applications.* Prentice Hall, Oct. 1983.

[10] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 784-803, Dec. 1997.

[11] D. Li and D.R. Cheriton, "OTERS (On-Tree Efficient Recovery Using Subcasting): A Reliable Multicast Protocol," *Proc. Int'l Conf. Network Protocols (ICNP '98)*, pp. 237-245, Oct. 1998.

[12] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proc. First Int'l Workshop Sensor Net Protocols and Applications*, pp. 345-353, Apr. 2003.

[13] C-Y. Wan, A. Campbell, and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," *Proc. ACM Int'l Workshop Sensor Networks and Architectures*, pp. 1-11, Sept. 2002.

[14] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicasting Routing Protocol," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '99)*, pp. 1298-1302, Sept. 1999.

[15] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," *Proc. ACM MobiCom*, Aug. 1999.

[16] D. Li and D.R. Cheriton, "Evaluating the Utility of FEC with Reliable Multicast," *Proc. Int'l Conf. Network Protocols (ICNP '99)*, pp. 97-105, Nov. 1999.

[17] W. Peng and X. Lu, "Efficient Broadcast in Mobile Ad Hoc Networks Using Connected Dominating Sets," *J. Software*, vol. 12, no. 4, pp. 529-536, Dec. 1999.

[18] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. ACM MobiHoc*, pp. 194-205, June 2002.

[19] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks," *Proc. ACM MobiHoc*, pp. 222-232, June 2003.

[20] O.B. Akan and I.F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 5, pp. 1003-1017, Oct. 2005.

[21] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm," *IEEE J. Selected Areas in Comm.*, special issue on ad hoc networks, vol. 17, no. 8, pp. 1454-1465, Aug. 1999.

[22] R.M. Karp, "Reducibility among Combinatorial Problems," *Complexity of Computer Computations*, no. 1, pp. 85-103, May 1972.

[23] T.P. Hayes, "Randomly Coloring Graphs of Girth at Least Five," *Proc. 35th ACM Symp. Theory of Computing (STOC '03)*, pp. 269-278, June 2003.

[24] V.V. Vazirani, *Approximation Algorithms.* Springer, May 2001.
[25] E.-S. Jung and N.H. Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks," *Wireless Networks,* vol. 11, pp. 55-66, 2005.
[26] *IEEE Standard 802 Part 15.4: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS),* IEEE, May 2003.
[27] A. Savvides and M.B. Srivastava, "A Distributed Computation Platform for Wireless Embedded Sensing," *Proc. 20th Int'l Conf. Computer Design (ICCD '02),* 2002.

Seung-Jong Park received the BS degree in computer science from Korea University, Seoul, in 1993, the MS degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Teajon, Korea, in 1995, and the PhD degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, in 2004. He is an assistant professor in the Computer Science Department and Center for Computation Technology, Louisiana State University. From 1995 to 2000, he had worked for Shinsegi Telecomm, which is the first code division multiple access (CDMA) cellular service provider in the world and has now merged with SK Telecom. He is a member of the IEEE.

Ramanuja Vedantham received the BTech degree in electrical engineering from the Indian Institute of Technology, Madras, in 2000, where he was the recipient of several scholarships and awards, the MS degree in computer science from the University of Texas at Austin in 2002, and the PhD degree in wireless networks from the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is a systems engineer at the Digital Signal Processing Solutions (DSPS) R&D Center, Texas Instruments, Dallas. He has published in leading conferences and journals such as ACM MobiCom 2006, ACM MobiHoc 2004, the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '06), the Second International Conference on Broadband Networks (BroadNets '05), the IEEE International Conference on Communications (ICC '05), the First International Conference on Wireless Internet (WICON '05), the *Ad Hoc Networks Journal*, and the *Computer Communications Journal*. He has been on the program committee of the IFIP Networking 2007 conference and has reviewed for several leading conferences and journals. His research interests are in the areas of wireless broadband networks, wireless sensor and actor networks, and protocols for energy-efficient operation for sensor and ad hoc networks. He is a member of the IEEE and the IEEE Communications Society.

Raghupathy Sivakumar received the BE degree in computer science from Anna University, Chennai, in 1996 and the PhD and MS degrees in computer science from the University of Illinois, Urbana-Champaign, in 2000 and 1998, respectively. He is an associate professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech). He leads the Georgia Tech Networking and Mobile Computing (GNAN) Research Group, where he and his students do research in the areas of wireless networking, mobile computing, and computer networks. He is a senior member of the IEEE.

Ian F. Akyildiz received the BS, MS, and PhD degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively. Currently, he is the Ken Byers distinguished chair professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, and the director of the Broadband and Wireless Networking Laboratory. He is the editor-in-chief of the *Computer Networks Journal*, as well as the founding editor in chief of the *Ad Hoc Networks Journal.* His current research interests are in next generation wireless networks, sensor networks, and wireless mesh networks. He received the "Don Federico Santa Maria Medal" for his services to the Universidad of Federico Santa Maria in 1986. From 1989 to 1998, he served as a national lecturer for ACM and received the ACM Outstanding Distinguished Lecturer Award in 1994. He received the 1997 IEEE Leonard G. Abraham Prize Award from the IEEE Communications Society for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures," published in the *IEEE Journal Selected Areas in Communications* in January 1996. He received the 2002 IEEE Harry M. Goode Memorial Award from the IEEE Computer Society with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He received the 2003 IEEE Best Tutorial Award from the IEEE Communication Society for his paper entitled "A Survey on Sensor Networks," published in the *IEEE Communications Magazine* in August 2002. He also received the 2003 ACM SIGMOBILE Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks." He received the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999 and 2003." He also received the 2005 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Institute of Technology. He has been a fellow of the ACM since 1996 and is also a fellow of the IEEE.