Design of Structure-Free and Energy-Balanced Data Aggregation in Wireless Sensor Networks

Chih-Min Chao[†] and Tzu-Ying Hsiao[‡]

[†]Department of Computer Science and Engineering
National Taiwan Ocean University, Taiwan
Email: cmchao@ntou.edu.tw (corresponding author)

[‡]Department of Computer Science and Engineering
National Taiwan Ocean University, Taiwan
Email: M96570002@ntou.edu.tw

Abstract

Power saving is a critical issue in wireless sensor networks since sensor nodes are battery-powered. Data aggregation is an effective approach to save energy because the number of transmissions can be reduced after aggregation. In the literature, most of the data aggregation protocols rely on a structured architecture to accomplish data gathering. Such structure-based methods suffer from high maintenance overhead in a dynamic environment where sensor nodes may move or fail unexpectedly. In this paper, we propose a structure-free and energy-balanced data aggregation protocol, SFEB. SFEB features both efficient data gathering and balanced energy consumption, which results from its two-phase aggregation process and the dynamic aggregator selection mechanism. Analysis, extensive simulation, and real system implementation results verify the superiority of the proposed mechanism.

Keywords: wireless sensor networks, structure-free data aggregation, energy efficient protocols

I. Introduction

A wireless sensor network (WSN) consists of numerous sensor nodes, each capable of collecting, processing, storing, and transferring environment information. Sensor nodes, normally deployed in an ad hoc manner, operate in a distributed way and coordinate with each other to fulfill a common task. A typical data reporting operation for WSNs relies on the cooperation of numerous sensor nodes and a sink node, which acts as a data collector. Due to the convenience and low cost of the sensor nodes, multiple applications of WSNs have been developed, including medical treatment, atmospheric monitoring, and traffic control. Sensor nodes have reduced computation and communication capabilities and are usually unrechargeable. Depending on the application, there are several kinds of reporting mechanisms for WSNs, such as periodical reporting, reporting by query, and event-driven reporting. In this paper, we focus on

A preliminary version of this paper has appeared in IEEE HPCC, 2009.

the event-driven reporting, where sensor nodes report their sensory data to the sink whenever a particular event occurs.

Power saving is an important issue for energy-constrained sensors. Numerous proposals for better power efficiency have been suggested. Some allow sensor nodes to enter power saving mode [1], [2], [3], some design new node placement approaches [4], [5], and some others try to reduce the number of transmissions [6], [7]. In this paper, we concentrate on a data aggregation mechanism that reduces energy consumption by lowering the number of transmissions. In the literature, data aggregation protocols could be classified into two classes: structured and structure-free. Structured solutions use a tree-based or a cluster-based structure constructed at the network initialization phase to achieve efficient data gathering. Such structured mechanisms perform well in a stable environment where nodes function properly at all times. However, in a more practical environment where nodes may move or fail unexpectedly, the benefit from structured gathering may not compensate for the necessary construction and maintenance overhead. On the contrary, structure-free approaches do not spend energy on building any structure. The Data-Aware Anycast and Randomized Waiting protocol (DAA+RW) [8] is a solution belonging to this class. It has been shown that, when compared with structured ones, structure-free solutions have the benefits of reduced average delay, reduced maintenance overhead, and better robustness when incurring node failures. Under certain circumstances, such as when the aggregation efficiency is low or when the network density is high, the structure-free solutions achieve even better performance [8].

Early aggregation is beneficial for saving energy since it reduces the number of packet transmissions. In DAA+RW, early aggregation is achieved only when a node that is closer to the sink selects a longer waiting time. To achieve more early aggregations, we utilize the “gather before transmit” concept. That is, certain data collecting sensor nodes (aggregators) are first selected to gather their neighbors’ sensory data. Then, these aggregators send the gathered local data back to the sink for final aggregation. To make this idea work, we have to answer the following two questions: “How many aggregators should be selected?” and “How long should these aggregators spend on gathering their neighbors’ data?” To conserve energy, the number of selected nodes should be as small as possible while still covering the event range and the duration for data collection should be as short as possible. In this paper, we propose a structure-free data aggregation mechanism to answer these two questions. Our protocol consists of two phases. In phase one, as an event occurs, data aggregators are dynamically selected to collect sensory data from their neighbors. In phase two, the aggregators send the collected information to the sink. The main contributions of the proposed solution can be summarized as follows: (1) Avoiding extensive packet exchanges, we design a lightweight aggregator election algorithm. Unlike other general leader/cluster head election methods, the specific aggregator election mechanism enables faster and efficient data aggregation. It also allows for balanced power consumption among nodes. (2) In phase two, based on DAA+RW, we propose an enhanced structure-free data gathering routing mechanism to further improve the aggregation effect (3) A mathematical analysis model is provided to verify the superiority of SFEB when nodes may move or fail unexpected. Analysis, simulation results, and real system implementation show that significant improvement on aggregation efficiency can be found when compared to the DAA+RW protocol, with small increase in delay in some situations.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Our protocol is described in Section 3. Performance analysis of the proposed protocol is in Section 4. Simulation and implementation results are shown in Section 5 and Section 6, respectively. Finally, conclusion remarks are drawn in Section 7.

II. Related Work

In this section, we review some existing data aggregation protocols according to whether a structure is used or not.

A. Structured Data Aggregation

Clustering is essential in cluster-based data gathering protocols. The protocol called Low-Energy Adaptive Clustering Hierarchy (LEACH) [6] is the first clustering algorithm in WSNs. In the set-up phase, a node running LEACH randomly determines if it is a cluster head. Communications between the sink and the cluster heads are based on CDMA, and thus, multiple cluster heads can transmit to the sink simultaneously. Each cluster head utilizes TDMA to schedule the transmissions among its cluster members. Many improvements of LEACH can be found in the literature [9], [10], [11], [12]. Two such schemes reduced overall energy consumption by considering sensor nodes' residual energy for cluster head selection [9], [10]. The HEED protocol [11] further improved the algorithm by considering the relative positions among nodes to avoid two sensor nodes that can communicate with each other becoming cluster heads concurrently. Another cluster head selection mechanism minimized the energy consumption [12]. A hierarchical data aggregation strategy has been proposed to reduce cluster head selection probability around the sink [14]. Another cluster-based data aggregation method [13] uses a tree structure as the transmission path for cluster members to their cluster heads and for cluster heads to the sink. Part of our scheme can also be regarded as a cluster head election algorithm. However, the cluster head selection algorithms mentioned above are not enough for data aggregation. Some important issues, such as reducing data collection interference and shortening the aggregating time, are not addressed. For the same reasons, the connected dominating set algorithms are not suitable for our problem.

Many other structures, such as chain-based or tree-based ones, were also proposed recently [7], [15], [16], [17]. In the Power-Efficient GAthering in Sensor Information Systems (PEGASIS), all nodes are organized into a chain and take turns communicating with the sink [15]. Another solution [7] suggested constructing an optimal multicast data aggregation tree such that any transmission can go through the least number of relay nodes in the tree; several mechanisms, including a greedy algorithm, an approximation algorithm, and a distributed approximation algorithm, were proposed. Liao et al. [16] constructed a data aggregation tree by an ant colony algorithm. Another tree-based method [17] suggested adaptive reconstructing the aggregation tree for different types of data. A common issue of these structured data aggregation schemes is the structure maintenance overhead, which greatly increases power consumption.

B. Structure-Free Data Aggregation

DAA+RW [8] is the first structure-free data aggregation scheme for event-driven reporting in sensor networks. Since there is no established data gathering structure, each node with event data to report sends an anycast RTS first to determine the next hop to the sink. Any node that receives this RTS is a next hop candidate. To achieve greater aggregation efficiency, a node that has the same event data to report or is closer to the sink has higher priority to respond a CTS. To reduce the number of transmissions, a randomized waiting scheme is introduced. Each node that has data to report can start its transmission after a random waiting time. Possible aggregation is generated when a node close to the sink chooses a longer waiting time. The main advantage of DAA+RW is that it avoids structure maintenance overhead. However, the randomized waiting time mechanism leaves rooms for improvement, as aggregation efficiency will be poor if nodes that are closer to the sink have selected shorter waiting times.

III. Protocol Description

We propose a structure-free and energy-balanced data aggregation protocol (SFEB). We assume that SFEB operates in a multihop network where sensor nodes are synchronized and are aware of the locations of the sink and their own. Location information can be obtained by applying a localization protocol that is either GPS-based [18], [19], [20], [21] or non-GPS-based [22], [23], [24], [25]. We also assume that nodes are randomly deployed and node density is available for all nodes. Each node has similar transmission and carrier sensing range. Packets with the same event identification (EID) can be aggregated.

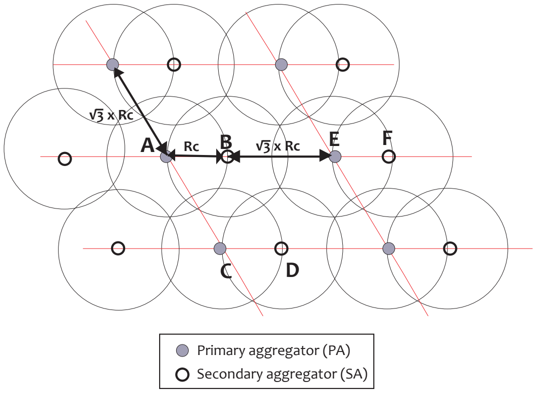
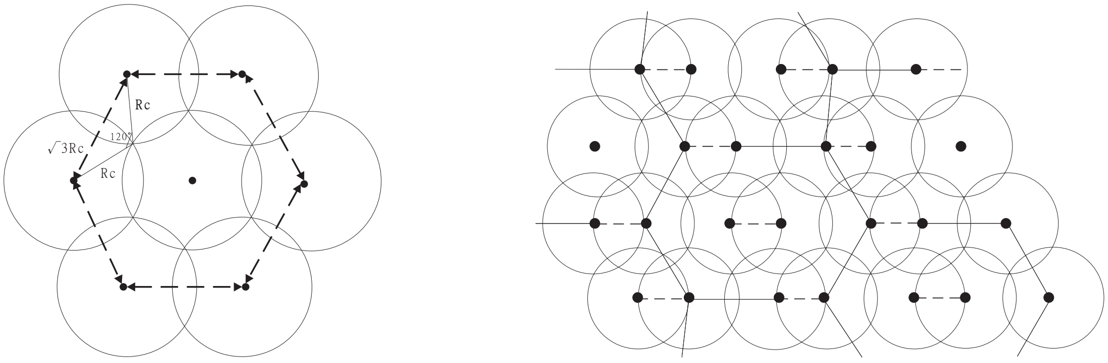


Fig. 1: Virtual parallelograms partitioning



(a)

(b)

Fig. 2: (a) Optimal node placement is in a hexagon pattern (b) Virtual parallelograms become virtual hexagons when PA and SA are considered as a virtual aggregated node

Our SFEB consists of two phases. In phase one, we designate some nodes as aggregators to gather as many packets as possible. Then, these aggregators send the collected packets back to the sink in phase two. Sensor nodes that fail to send data to aggregators will also transmit their packets to the sink in phase two.

A. *SFEB Phase One*

The first task in phase one is to select the aggregators. To find the aggregators, the network is partitioned into virtual parallelograms as shown in Fig. 1. The side widths of a parallelogram are $(1+\sqrt{3})R_c$ and $\sqrt{3}R_c$, where R_c is the communication range of sensor nodes. In the ideal case, there always exist some nodes located at the parallelogram corners. These nodes are designated as the primary aggregators (PAs). For example, in Fig. 1, nodes A, C, and E are PAs. Associated with each PA there will be a secondary aggregator (SA) located at R_c away. Without loss of generality, we assume the SAs are on the right of PAs. In Fig. 1, nodes B, D, and F are SAs of nodes A, C, and E, respectively. Here two aggregators, a

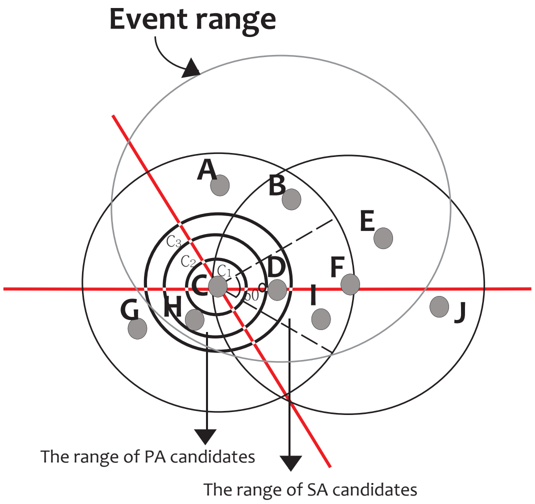


Fig. 3: The contention ranges of a PA and an SA

PA and an SA, are regarded as a pair for the next step. When an aggregator finishes collecting data, it has to find a node to forward its data back to the sink. For each PA/SA pair, we designate the aggregator that is closer to the sink as the forwarding node of the other. Such a mechanism accomplishes both early aggregation and the first forwarding node selection. To make the number of aggregators low while keeping the event range full-covered, the ideal positions of different pairs of PA and SA are $\sqrt{3}\ R_c$ away. The is based on the known fact that, for the 1-coverage problem, placing homogeneous sensors in a hexagon pattern is optimal [26]. As shown in Fig. 2(a), the distance between any two sensor nodes is $\sqrt{3}\ R_c$ in the hexagon pattern. To facilitate our presentation, we use virtual parallelograms; however, if we consider a PA/SA pair as a virtual aggregated node, the virtual parallelograms become virtual hexagons as shown in Fig. 2(b).

Earlier we assume that there exist some nodes at the parallelogram corners; however, it is not the case in reality. To solve this problem, nodes contend to act as PAs in SFEB. When an event occurs, nodes detecting this event with their residual energy higher than a predefined threshold and their distances to any parallelogram corner less than $\frac{1}{2}R_c$ are PA candidates. To contend to be a PA, these nodes compete to broadcast a PA Request (PAR) packet where the location of the sender is included. We let the one that first sends a PAR become a PA. To keep the selected PA close to the parallelogram corner, we divide the PA candidates into three equal-width coronas, C_1, C_2, and C_3, as shown in Fig. 3, where candidates in C_1 have the highest priority and those in C_3 have the lowest. Specifically, the contention window for PA selection, denoted as CW_{PAR}, is partitioned into three parts. The length of each part may be different and is denoted as CW_{PAR_1}, CW_{PAR_2}, and CW_{PAR_3}. Nodes located in C_1, C_2, and C_3 choose their backoff time within the range of 0 to CW_{PAR_1}, CW_{PAR_1}+1 to CW_{PAR_1}+CW_{PAR_2}, and CW_{PAR_1}+CW_{PAR_2}+1 to

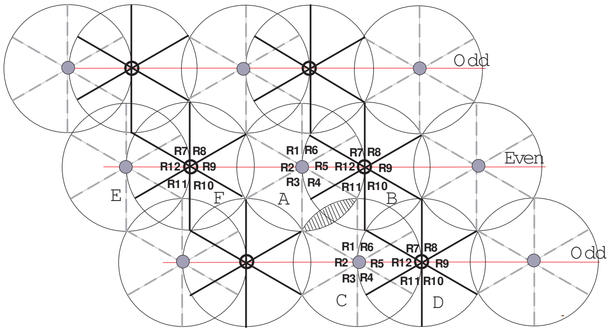


Fig. 4: The 12 subregions of a PA/SA pair

CW_{PAR}, respectively.

The length of CW_{PAR_i}, $i = 1, 2, 3$, can be estimated based on the work of Hsu et al. [27], where the expected number of successful transmissions for n contenders with contention window size of m_i, denoted as $SUC(n, m)$, is derived. In our PA selection scenario, at least one PAR transmission must be successful for the PA selection. Thus, when the number of PA candidates in corona C_i is n_i, which is estimated from the network density, CW_{PAR_i} is set to m_i such that $SUC(n_i, m_i) \geq 1$.

After a PAR is received, the SA candidates, defined as nodes located within the SA contention region with their residual energy higher than a predefined threshold, contend to respond with an SA Request (SAR) packet to act as the corresponding SA. The calculation to obtain the appropriate contention window can be applied only if the SA contention region is a single hop one [27]. We define this region to be a fan-shaped area on the right of the associated PA, as shown in Fig. 3. It should be noted that a fan-shape area is constructed according to the position of each selected PA instead of the intersection point of the virtual parallelograms. The contention window that minimizes the SA contention period, denoted as CW_{SAR}, can be determined in a similar way as the CW_{PAR}. The duration for PA and SA selection, $T_{selection}$, can be obtained by

$$\begin{aligned} T_{selection} = CW_{PAR} \times slot\_time + TransTime(PAR) + \\ CW_{SAR} \times slot\_time + TransTime(SAR) \end{aligned}$$

where $slot\_time$ is the length of a time slot and $TransTime(X)$ is the time for transmitting the packet X.

It should be noted that sensory data of any non-aggregator node will be aggregated by a single aggregator. Also, a node will not send its sensory data at this phase if neither PAR nor SAR is correctly received. When a PAR (or SAR) is received, a non-aggregator node that has detected the same event will send its packet to the PA (or SA). That is, the next job is for the selected aggregators to collect their neighbors' sensory data. The problem to be solved here is how long should the PAs and SAs spend on collecting sensory data. The duration of sensory data collection for each aggregator is determined by the number of their neighbors. The contention period calculation mechanism proposed by Hsu et al. may be a solution but it only applies to a single hop environment [27]. Thus, we evenly partition the coverage area of PA into six single hop fan-shaped subregions. Such a partition makes the minimum number of single-hop subregions. The coverage area of the associated SA is also evenly partitioned and thus a total of 12 fan-shaped subregions, $R1$ to $R12$, are partitioned for each PA/SA pair as shown in Fig. 4. The optimal duration for data gathering of each subregion can thus be estimated. To facilitate our description, the aggregators are also divided horizontally into two different categories, odd and even, as shown in Fig. 4. Non-aggregator nodes know which subregion they belong to since they have the PA's or the SA's location information.

Each aggregator has to gather sensory data from several regions. It is not trivial to determine the sequence of these subregions. A simple method that allows all PA/SA pairs to collect concurrently from $R1$ to $R12$ is infeasible. For example, consider the pairs A/B and C/D in Fig. 4 when collecting their subregion $R1$. The transmissions from nodes in the shaded area and those from nodes in subregion $R1$ of pair A/B may collide at A. To schedule a feasible aggregation sequence, we propose the following scheduling principle.

Principle 1. *Subregions that can be aggregated concurrently will be scheduled first, if the following conditions hold.*

1. Nodes within the subregions will not produce collisions.

2. The time of concurrent aggregation is shorter than that of separate aggregation.

For example, $R2$ and $R9$ in each aggregator pair can be aggregated at the same time to associated PA and SA, respectively. Typically, an SA is outside the transmission range of a node in $R2$ and thus collisions with nodes in $R9$ of the same aggregation pair are avoided. In addition, collisions can be avoided at the neighboring PA/SA when the carrier sensing is applied. For $R2$ and $R9$, the concurrent aggregation time is shorter than the sum of separate aggregation times because only a small part of the nodes in a subregion may interfere with nodes in the other subregion.

To reduce the length of aggregation period, it is desirable to find as many subregions as possible that satisfy Principle 1. We first try to find the subregions in the same aggregator pair that can be scheduled simultaneously. Since an aggregator cannot collect two subregions without collision at the same time, we list the concurrent aggregation feasibilities of a PA/SA pair in Tab. I. Most subregions are unable to

TABLE I: Concurrent aggregation feasibilities for subregions in a PA/SA pair

	R1	R2	R3	R4	R5	R6
R7	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)
R8	No(CD)	No(CD)	Yes	No(CS)	No(CS)	No(CS)
R9	No(CD)	Yes	No(CD)	No(CS)	No(CS)	No(CS)
R10	No(CD)	No(CD)	No(CD)	No(CS)	No(CS)	No(CS)
R11	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)
R12	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)	No(CS)

TABLE II: Concurrent aggregation feasibilities for combinations of subregions in different PA/SA pairs

		Odd						
		R1	R4	R5	R6	R7	R10	R11
	R1	No(CD)	Yes	No(CD)	No(CD)	No(CD)	No(CD)	No(CD)
	R4	Yes	No(CD)	No(CD)	No(CD)	No(CD)	No(CD)	No(CD)
	R5	No(CD)	No(CD)	Yes	No(IF)	No(IF)	No(CD)	No(CD)
Even	R6	No(CD)	No(CD)	No(IF)	Yes	No(CD)	No(CD)	No(CD)
	R7	No(CD)	No(CD)	No(IF)	No(CD)	No(CD)	Yes	No(CD)
	R10	No(CD)	No(CD)	No(CD)	No(CD)	Yes	No(CD)	No(CD)
	R11	No(CD)	No(CD)	No(CD)	No(CD)	No(CD)	No(CD)	Yes

schedule concurrently because of possible collisions from transmissions in the same pair (denoted as CS) or in a different pair (denoted as CD). The only subregion pairs that can be gathered together are $R2/R9$ and $R3/R8$.

Concurrent aggregation in different aggregator pairs for the rest of the subregions is also possible. Among them, subregions $R5$ or $R12$ can be covered if the other 11 subregions have been aggregated. Without loss of generality, $R12$ is not considered in our scheduling. Tab. II lists the feasibilities of 49 different combinations for the subregions of odd and even aggregators. Most combinations cannot be scheduled concurrently due to possible collisions or lack of reduction in aggregation time because of interference (denoted as IF). For those subregion pairs that can be gathered at the same time, each subregion exists exactly in one combination. This means that the aggregation sequence can be determined randomly. Without loss of generality, the final aggregation schedule for an aggregator pair is shown in Tab. III.

A total of 9 durations are needed to gather data from neighboring nodes. The notations O$(\cdot)$ and E$(\cdot)$ represent scheduled subregions in odd and even aggregators, respectively. The area of an aggregator's transmission range is represented by A_{R_C}. The expected number of contending nodes for a duration i, denoted as NC_i, in a subregion is the product of the network density D and the uncollected area of

TABLE III: The final aggregation schedule

Duration	PA	SA	Expected number of contending nodes
1	R2	R9	$\frac{1}{6}\times A_{R_C}\times D$
2	R3	R8	$\frac{1}{6}\times A_{R_C}\times D$
3	O[R1]+E[R4]	-	$\frac{1}{6}\times A_{R_C}\times D$
4	O[R4]+E[R1]	-	$\frac{1}{6}\times A_{R_C}\times D$
5	-	O[R7]+E[R10]	$\frac{1}{6}\times A_{R_C}\times D$
6	-	O[R10]+E[R7]	$\frac{1}{6}\times A_{R_C}\times D$
7	R5	-	$\frac{1}{8}\times A_{R_C}\times D$
8	R6	-	$\frac{1}{18}\times A_{R_C}\times D$
9	-	R11	$\frac{1}{24}\times A_{R_C}\times D$

the subregion. This value is essential for calculating the optimal contention window for each duration i, denoted as CW_{D_i}. The contention window for NC_i neighboring nodes that can minimize the aggregation period can be obtained as follows [27].

$$CW_{D_i} = \arg \min_{m \in 2^i, i \in N} \frac{T_{cp}}{SUC(NC_i, m)}$$

where T_{cp} is the length of the contention period. A predefined additional duration, EIT_i, is appended for durations 1 to 8 because of the increased interference nodes. The length of duration i, denoted as T_i, is obtained as follows.

$$Ti = \begin{cases} CW_{D_i} \times slot\_time + TransTime(Data) \times NC_i + EIT_i, & \text{if } i \leq 8 \\ CW_{D_i} \times slot\_time + TransTime(Data) \times NC_i, & \text{otherwise.} \end{cases}$$

The total time for a PA/SA pair to collect sensory data, denoted as $T_{aggregation}$, is given by

$$T_{aggregation} = \sum_{i=1}^{9} T_i$$

For those nodes that are neighbors of a PA or SA, sensory data are sent to their aggregator using IEEE 802.11 DCF during the $T_{aggregation}$ time span. The exact reporting time span depends on the subregion in which these nodes belong to. Nodes located on the intersections of two different regions belong to both regions. In such cases, these nodes will transmit with the earlier of the two subregions. For example, a node that belongs to both $R6$ and $R7$ will transmit at $R7$ since $R7$ is scheduled earlier.

Once a PA/SA finishes data collection, it will broadcast an acknowledgement (ACK) packet. An ACK packet contains the identifications of the sensor nodes that have successfully transmit their data. Without loss of generality, we assume that the SA has longer distance to the sink in each PA/SA pair. In each PA/SA pair, the SA broadcasts an ACK first. Then, the collected data from the SA is delivered to the PA. Lastly, the PA broadcasts an ACK packet if data are received. To avoid collision, odd/even aggregators have separate ACK packet transmissions. The length of ACK transmission duration, denoted as T_{ACK}, is obtained by

$$T_{ACK} = 2 \times (2 \times TransTime(ACK) + TransTime(Data))$$

Although the process of SFEB phase one involves complex computing, the communication overhead of sensor nodes in this phase is low. The overhead includes PAR/SAR transmissions for non-aggregator nodes and extra ACK transmissions for aggregators. The PAR/SAR packets also provide the functionality of RTS/CTS. We comment that SFEB phase one achieves energy efficient clustering and fast aggregation with little control overhead[1].

B. SFEB Phase Two

The aggregators can gather all the sensory data in their vicinity if SFEB phase one works perfectly. The remaining job is to send the collected data back to the sink. However, some sensor nodes may fail to transmit data to PAs or SAs at phase one (these nodes are denoted as orphan nodes[2]). These orphan nodes also have to send their data to the sink at this phase. Since any node may become an orphan node, all nodes within the event range must know when the SFEB phase two begins for synchronized transmission. Fortunately, the duration of SFEB phase one is easy to obtain: it is the summation of the aggregator selection time ($T_{selection}$), the data collecting time ($T_{aggregation}$), and the ACK transmission time (T_{ACK}).

To achieve better aggregation, rear aggregators (RAs) that are at the rear side of the event range should transmit as early as possible. For example, in Fig. 5, it is obvious that nodes A, B, C, and D should report earlier than nodes F and H. To identify an RA, rear subregions must be defined first. In the six subregions of a PA or an SA, the three rear subregions are the ones that have longer distances to the sink. We define six different cases according to the angle from the PA/SA to the sink as shown in Fig. 6, where shaded half are rear subregions. In SFEB phase one, the number of collected packets in each subregions is recorded. An aggregator is considered as an RA if this number is less than a predefined

[1] For PAR, the added information include EID, PA's location, and the class (even or odd) the source node belongs to. For SAR, the added information include EID, SA's address and location, PA's ID, and the class the source node bolongs to. For ACK, the added fields include the IDs of the sensor nodes that have been successfully received the packet sent by the aggregator.

[2] Orphan nodes include these sensor nodes that are not covered by any aggregators or do not receive the PAR or SAR packet successfully.

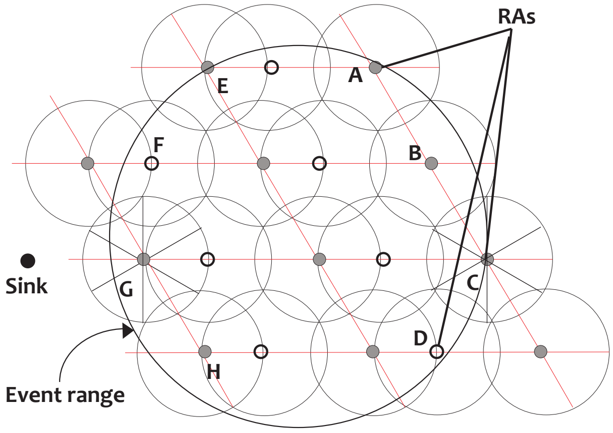


Fig. 5: The RAs

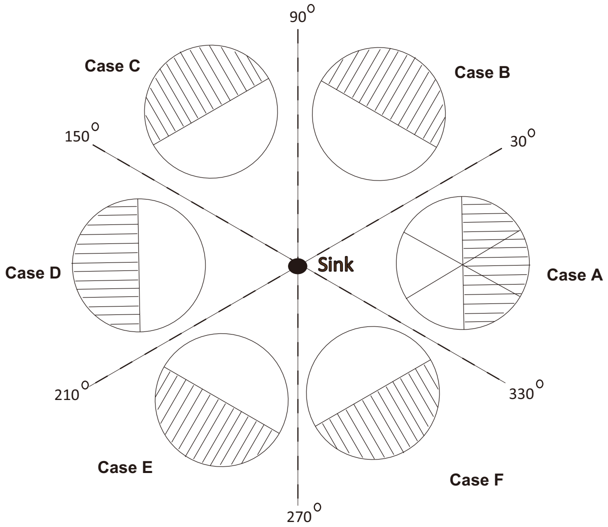


Fig. 6: The rear subregion classification

threshold for at least two of its rear subregions. Take Fig. 5 as an example again, the nodes A, C, and D are likely to be RAs. In SFEB, the RAs report to the sink immediately at the beginning of SFEB phase two. The other aggregators and the orphan nodes, following the DAA+RW scheme, wait a random time before reporting their data.

Classifying CTS priorities in DAA+RW can improve reporting efficiency. In this paper, we revise the priority mechanism to further improve the reporting efficiency. In DAA+RW, most nodes are allowed to respond a CTS to any RTS. It is possible for a low-priority node to act as a forwarding node even though high-priority nodes are available. To increase the probability of high-priority nodes becoming forwarding nodes, we consider the number of times that the RTS packet has been transmitted. That is, when an RTS packet is transmitted for the first time, only nodes with the highest priority can respond a CTS. When an RTS packet is transmitted for the second time, nodes with the second highest priority can also respond a

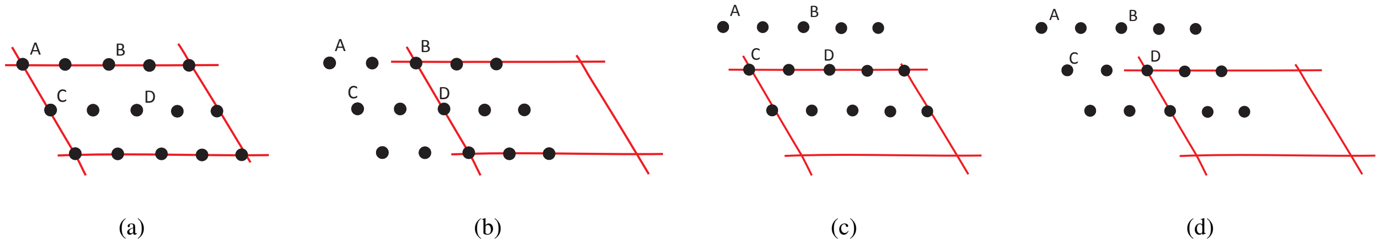


Fig. 7: An example of virtual parallelograms movement with S_p = 2 and S_d = 10 at time (a) 0 $\sim$ 9 seconds (b) 10 $\sim$ 19 seconds (c) 20 $\sim$ 29 seconds (d) 30 $\sim$ 39 seconds

CTS. Following this method, we insure that higher priority nodes become forwarding nodes. This modified SFEB mechanism with CTS reply limitation is denoted as SFEB-CL.

C. *Energy-Balanced Mechanism*

Sharing the data gathering duty among nodes is necessary because PAs and SAs consume more energy. To achieve this goal, the virtual parallelograms used in SFEB is moved periodically. Two parameters, S_p and S_d, are needed in our mechanism. The former is the number of positions the virtual parallelograms can change vertically or horizontally. The latter is the dwell time for each movement. Both parameters could be set according to the R_c and network density during network initialization. Fig. 7 is an example of virtual parallelograms movement with S_p = 2 and S_d = 10 seconds. As can be seen in the figure, different nodes may act as the PA over time.

D. *Complexity*

The complexity of the SFEB is calculated as follows. The main tasks in SFEB phase one include PA selection, SA selection, data aggregation, and ACK broadcasting. In the first three tasks, the computation time is dominated by the process of calculating proper contention window size, which takes a complexity of O(cslog$s$) where c is the number of contenders and s is the least expected number of successful transmissions. PA selection and SA selection need only one successful transmission ($s = 1$) and thus the complexity are both O($c$log$c$). Data aggregation requires $s = c$ which means the complexity is O(c^2log$c$). For the ACK broadcasting, the complexity is O(1). The complexity for SFEB phase two is also O(1), which makes the time complexity of SFEB O(c^2log$c$).

IV. Performance Analysis

We built an analytical model to estimate the total number of transmitted packets when different structure-free data aggregation protocols, SFEB and DAA+RW, are used. An ideal tree-structured data

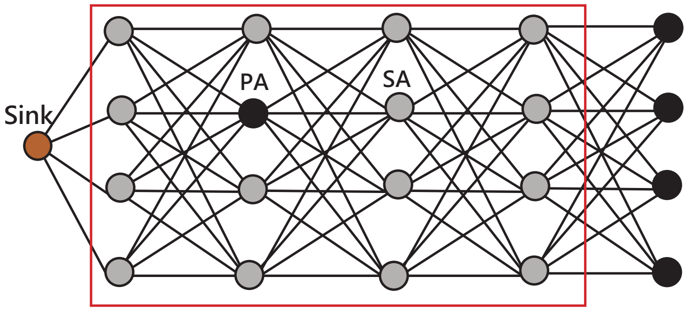


Fig. 8: An example of network topology

aggregation protocol (TSDA) is also calculated for comparison purposes. To clarify the effectiveness of early aggregation, we consider control and data packets separately. We assume that all the nodes have packets to send. To simplify our analysis, we consider the packet transmission delay to be zero and the control packet transmission to be collision-free. Also, the RA selection in SFEB phase two is not considered. In our analysis, n sensor nodes are distributed in the network where each one has k downstream nodes and k upstream nodes. A node is only capable of communicating with its upstream and downstream nodes. An example network topology of n = 20 and k = 4 is shown in Fig. 8.

First, we analyze the TSDA protocol. In TSDA, there exists a data gathering tree rooted at the sink. Data aggregation starts from the leaf nodes after successful RTS/CTS exchanges. The intermediate nodes transmit their packets after collecting all the data from their children. With a data gathering tree, the number of transmissions for data and control packets in the data aggregation process is n and $2n$, respectively.

For the DAA+RW mechanism the number of transmissions for data packet is $(n+1)H_k(\frac{n}{k}) - \frac{n}{k}$, where $H_k(n) = \sum_{i=1}^{n} \frac{1}{(i-1)k+1}$ is the summation of a harmonic sequence [8]. In DAA+RW, each data transmission is followed by a successful exchange of RTS and CTS packets. That is, the number of transmissions for control packets is two times of that of data packets.

The SFEB protocol consists of two phases. In phase one, PAs and SAs are selected. In our analysis, we assume that the PA is closer to the sink than the SA in each PA/SA pair. Consider the topology in Fig. 8 again. Without loss of generality, the PA and SA are marked explicitly. The nodes in the data gathering range of each pair of PA and SA form an *aggregation set* which is indicated by a rectangle. A complete aggregation set consists of $4k$ nodes. It should be noted that n is not always dividable by $4k$. Let r be the remainder of n divided by $4k$ (r = n mod $4k$). According to the value of r, we calculate the number of transmissions from the following four cases:

- Case A: r = 0, which means that the network consists of complete aggregation sets.
- Case B: $0 < r \leq k$, which means that these r nodes cannot form any complete or incomplete

aggregation set with an aggregator.

- Case C: $k < r \leq 2k$, which means that these r nodes form an incomplete aggregation set with a PA.
- Case D: $r < 4k$, which means that these r nodes form an incomplete aggregation set with a PA and an SA.

In Case A, we can see that there are $\frac{n}{4k}$ pairs of PA/SA while one PAR and one SAR must be exchanged for each pair. Thus, a total of $\frac{2n}{4k}$ control packets are transmitted. With the help of the corresponding SA, each PA is responsible for collecting data from $4k-1$ nodes (including the SA). This means the total number of data packet transmissions is $(4k-1)(\frac{n}{4k})$. In Cases B, C, and D, there are $\frac{n-r}{4k}$ complete aggregation sets. The r nodes in Cases C or D form an incomplete aggregation set. Thus, there will be one (PAR) or two (PAR and SAR) more control packets and some data packets to be sent for Cases C and D, respectively. It should be noted that there exist some nodes that fail to find an aggregator in Cases B and C. These nodes will transmit their packet back to the sink in phase two.

In SFEB phase two, nodes follow the DAA+RW mechanism to report data back to the sink. Therefore, the number of transmissions for control packets is two times of that of data packets. The data aggregation in SFEB phase one must be successful since we assume control packets are collision-free. Thus, nodes that have packets to send in the reporting of SFEB phase two include the PA in each aggregation set and the nodes that fail to find an aggregator to collect their data. To facilitate our calculation, we classify these nodes according to their hop count distance to the sink. A node belongs to the first class, denoted as CLS_1, has the hop count distance h, where $h \ mod$ 4 = 2. A node belongs to the second class, denoted as CLS_2, has the hop count distance of h, h = H and $H \ mod$ 4 = 1, where H is the maximum hop count distance in the network. In phase two, nodes belong to CLS_1 or CLS_2 follow the DAA+RW mechanism to report data back to the sink. This means that they will wait for random delays before their transmissions. However, nodes belonging to neither CLS_1 nor CLS_2, but also help relay packets, have no delay.

Next we calculate the number of data packet transmissions. Let Y be the random variable signifying the number of hops a packet has been forwarded when it is aggregated. When a node v_j sends a packet, the packet will be aggregated only at a node that has chosen a bigger random waiting time than that of v_j. That is, a packet is aggregated after being forwarded i hops means that the first $i-1$ relay nodes have zero or smaller random waiting times while the i-th node has a larger waiting time. For a packet sent by a node in the CLS_1 with h hops away from the sink, the probability that the packet is aggregated i hops away is

$$\text{P}(Y = i|h \ mod \ 4 = 2) \ = \begin{cases} x^{(\frac{i}{4}-1)} \cdot (1-x), & \text{if } i \ mod \ 4 = 0 \text{ and } 0 < i < h \\ x^{(\frac{i-2}{4})}, & \text{if } i = h \\ 0, & \text{otherwise.} \end{cases}$$

where x is the normalized random waiting time the node selected. The expected value of Y for a node with $RW = x$ is

$$\text{E}[Y|h \ mod \ 4 = 2, \ RW = x]$$

$$= \sum_{i \in S} i \cdot [x^{(\frac{i}{4}-1)} \cdot (1-x)] + h \cdot x^{(\frac{h-2}{4})}$$

$$= 4 \cdot \sum_{p=0}^{(\frac{h-2}{4}-1)} x^p + 2 \cdot x^{(\frac{h-2}{4})}$$

where S = [$i|4 \leq i \leq (h-2)$, $i \ mod$ 4 = 0]. Since the range of x is between 0 and 1, thus E[Y] is

$$\text{E}[Y|h \ mod \ 4 = 2]$$

$$= \int_0^1 \text{E}[Y|h \ mod \ 4 = 2, \ RW = x]dx$$

$$= \int_0^1 (4 \cdot \sum_{p=0}^{(\frac{h-2}{4}-1)} x^p + 2 \cdot x^{(\frac{h-2}{4})})dx$$

$$= 4 \cdot \sum_{p=0}^{(\frac{h-2}{4}-1)} \frac{1}{p+1} + \frac{8}{h+2} \tag{1}$$

Similar analysis can be applied to nodes in CLS_2. The probability that the packet is aggregated i hops away is given by

$$\text{P}(Y = i| \ h = H, \ H \ mod \ 4 = 1) \ = \begin{cases} x^{(\frac{i-3}{4})} \cdot (1-x), & \text{if } i \ mod \ 4 = 3 \text{ and } 0 < i < H \\ x^{(\frac{i-1}{4})}, & \text{if } i = H \\ 0, & \text{otherwise.} \end{cases}$$

The expected value of Y for a node with $RW = x$ is

$$\text{E}[Y|h = H, \ H \ mod \ 4 = 1, \ RW = x]$$

$$= \sum_{i \in S} i \cdot [x^{(\frac{i-3}{4})} \cdot (1-x)] + H \cdot x^{(\frac{H-1}{4})}$$

$$= 4 \cdot \sum_{p=0}^{(\frac{H-5}{4})} x^p + 2 \cdot x^{(\frac{H-1}{4})} - 1$$

where S = [$i|$ $3 \leq i \leq (H-2)$, i mod 4 = 3].

Finally, E[Y] is

$$\text{E}[Y|h\,=\,H,\;H\;mod\;4\,=\,1] = 4\cdot\sum_{p=0}^{(\frac{H-5}{4})}\frac{1}{p+1}+\frac{8}{H+3}-1 \tag{2}$$

With these expected values, we can obtain the expected number of transmissions for different cases. For Cases A and D, the expectation of Y is obtained from equation (1) because all nodes participating SFEB phase two in these two cases belong to CLS_1. The number of transmissions, denoted as $T_{2A,A}$ (= $T_{2A,D}$), is

$$T_{2A,A}\;\;=\;\;T_{2A,D}\;\;=\textstyle\sum_{h\in S}(4\cdot\sum_{p=0}^{\frac{h-2}{4}-1}\frac{1}{p+1}+\frac{8}{h+2})$$

where S = [$h|$ $1 \leq h \leq H$, h mod 4 = 2].

For Case B, some nodes participating SFEB phase two belong to CLS_1 and the others, locating at the last column of the incomplete aggregation set, belong to CLS_2. The expectation for Case B, denoted as $T_{2A,B}$, obtained from equations (1) and (2) is

$$T_{2A,B}\;\;=\textstyle\sum_{h\in S}(4\cdot\sum_{p=0}^{\frac{h-2}{4}-1}\frac{1}{p+1}+\frac{8}{h+2})+r\cdot(4\cdot\sum_{p=0}^{\frac{H-5}{4}}\frac{1}{p+1}+\frac{8}{H+3}-1)$$

where S = [$h|$ $1 \leq h \leq (H-1)$, h mod 4 = 2].

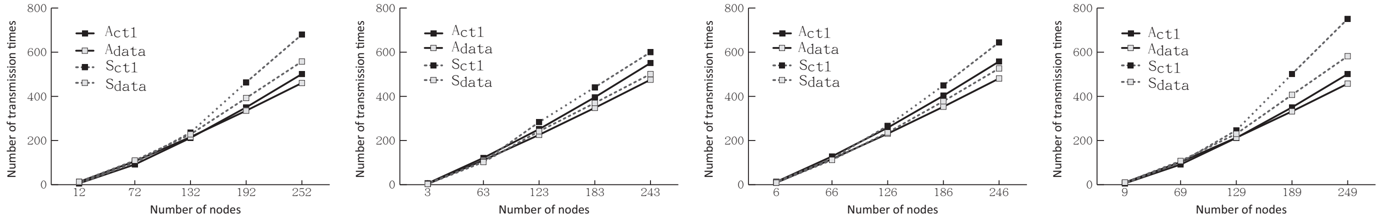
For Case C, since nodes in the same column cannot communicate with each other, the nodes in the last column of the incomplete aggregation set will transmit at SFEB phase two. The expectation of the number of packet transmissions for nodes in Case C, denoted as $T_{2A,C}$, is

$$T_{2A,C}\;\;=\textstyle\sum_{h\in S}(4\cdot\sum_{p=0}^{\frac{h-2}{4}-1}\frac{1}{p+1}+\frac{8}{h+2})+(r-k)\cdot(4\cdot\sum_{p=0}^{\frac{H-2}{4}-1}\frac{1}{p+1}+\frac{8}{H+2})$$

where S = [$h|$ $1 \leq h \leq (H-2)$, h mod 4 = 2]. The number of transmissions for SFEB is summarized in Tab. IV.

TABLE IV: The number of transmissions for four cases in SFEB

	Control packets		Data packets	
Case	Phase one	Phase two	Phase one	Phase two
A	$2(\frac{n}{4k})$	$2(T_{2A,A})$	$(\frac{n}{4k})(4k-1)$	$T_{2A,A}$
B	$2(\frac{n-r}{4k})$	$2(T_{2A,B})$	$(\frac{n-r}{4k})(4k-1)$	$T_{2A,B}$
C	$2(\frac{n-r}{4k})+1$	$2(T_{2A,C})$	$(\frac{n-r}{4k})(4k-1)+k$	$T_{2A,C}$
D	$2(\frac{n-r}{4k})+2$	$2(T_{2A,D})$	$(\frac{n-r}{4k})(4k-1)+(r-1)$	$T_{2A,D}$



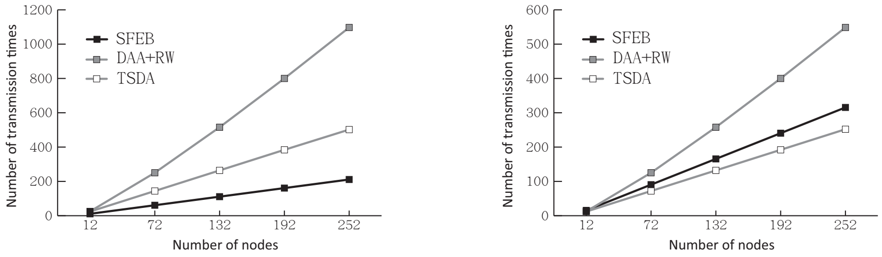
(a) (b) (c) (d)

Fig. 9: The transmission times for the analysis and simulation results for (a) case A (b) case B (c) case C (d) case D

To verify the accuracy of our analysis, we compare the analysis results with simulations using ns-2 (version ns-allinone-2.30). We set k = 3 and vary n for four cases. The value of n for Cases A, B, C, and D is $(20 \cdot j + 4)k$, $(20 \cdot j + 1)k$, $(20 \cdot j + 2)k$, and $(20 \cdot j + 3)k$, respectively, where j is an integer between 0 and 4. In Fig. 9, A_{ctl} and S_{ctl} represent the number of transmission attempts for the control packet obtained from analysis and simulation, respectively, while A_{data} and S_{data} represent the number of transmission attempts for the data packet obtained from analysis and simulation, respectively. As shown in Fig. 9, the results of analysis and simulation are close for all the cases when n is small. More deviation can be found when n increases due to the fact that collision and transmission time cannot be eliminated in the simulations. As n increases, the number of hops increases accordingly which enlarges the difference between the analysis and simulation models.

Next, we show the analysis results of TSDA, DAA+RW, and our SFEB when k = 3 and n = $(20 \cdot j + 4)k$, where j is an integer between 0 and 4. In Fig. 10(a), we can see that the SFEB reduces the number of control packet transmissions significantly when compared with DAA+RW. For example, when n = 253, the number of control packet transmissions for DAA+RW and SFEB is 1097 and 211, respectively. As shown in Fig. 10(b), the SFEB also reduces the number of data packet transmissions. The gap between SFEB and DAA+RW enlarges as n increases. This indicates that our protocol is more suitable for application to a large scale network.

It should be noted that it is difficult to tell whether TSDA performs better than SFEB according to Fig. 10. This is because i) we consider the number of control and data packets instead of the sizes of them and ii) the structure maintenance overhead (such as construction and reconstruction signaling) is not considered. In particular, structure maintenance overhead should be considered for better contrast of the two protocols. However, existing data aggregation protocols have different structured mechanism and thus, impose different overheads in these protocols. To make a more general comparison, we make the



(a) (b)

Fig. 10: The transmission times of different protocols for the analysis for the (a) control packet and (b) data packet

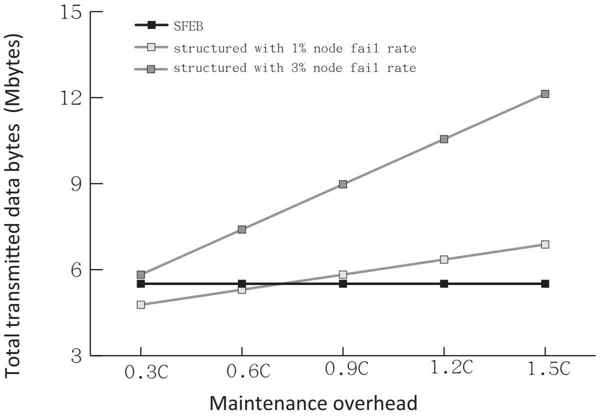


Fig. 11: Relation between transmitted data and maintenance overhead

structure overhead a variable and calculate the number of accumulated bytes transmitted for SFEB and TSDA. In this comparison, the network consists of 250 sensor nodes with k = 5. Only the rightmost 50 sensor nodes are within the event range. The packet size for RTS, CTS, and ACK is 20, 14, and 14 bytes, respectively. The ID and location fields are set to 1 and 2 bytes, respectively. This means that the packet size for PAR, SAR, and the ACK packet in SFEB phase one are 23, 24, and $(14+2\cdot5)$=24 bytes, respectively. The data packet is 70 bytes long. The structure maintenance overhead for each node is varied from $0.3C$ to $1.5C$, where C is the size of the shortest control packet. For example, C is set to 14 bytes in our analytical model. The node failure rate is set to 1% and 3% per minute.

Fig. 11 shows the relationship between total transmitted data and maintenance overhead for our SFEB and a structured protocol with different node failure rates. The results are collected 200 minutes after the first event is generated (an event is generated every 30 seconds). As expected, the total transmitted data of the structured protocol increases proportional to the maintenance overhead. Comparing with SFEB, the structured protocol with 1% node failure rate performs better when the maintenance overhead is less than $0.6C$. When node failure rate is increased to 3% per minute, SFEB will consistently offer better perfromance.

V. Simulation Results

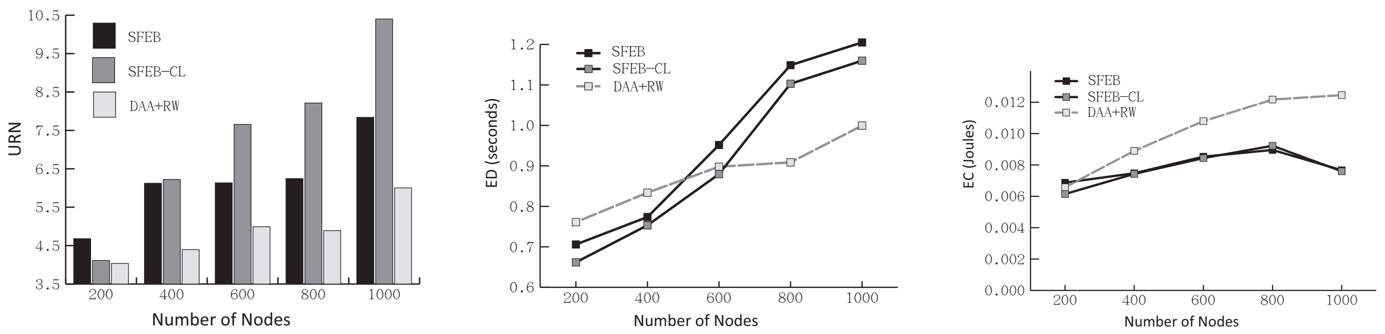
We have implemented a simulator using ns-2 (version ns-allinone-2.30) to evaluate the performance of the proposed protocols SFEB and SFEB-CL. We have also implemented DAA+RW for comparison purposes. Unless otherwise specified, 400 sensor nodes were randomly placed within a square area of side length 500 meters. The transmission range of a sensor node is 50 meters while the data rate is 100 kbps. Events are generated every 3 seconds with a radius of 40 meters. Each reported packet is 70 bytes long. Each sensor node has an initial energy of 70 Joules and the power consumption for transmit, receive, and idle modes is 0.081, 0.03, and 0.01 W, respectively. A spot in the following figures is the average of 20 simulation runs, each simulating 32 seconds (10 events). When a node with an event packet to send receives another packet with the same EID, the node merges the two packets into one. The size of the aggregated packet depends on the aggregation function which is given by $max\{original\_event\_packet\_size, N_{ep} \times (1 - \rho) \times original\_event\_packet\_size\}$, where N_{ep} is the number of effective packets and ρ is the aggregation ratio. Here ρ = 1 is perfect aggregation where the size of two merged packets remains the same as that of the original unmerged packet. In our simulations, the value of ρ is 1 if not otherwise specified. The threshold to determine if an aggregator is an RA is $\frac{A_{R_C} \times D}{24}$ while the EIT_i is set to 0 for all i.

Three metrics are observed in our simulations. The first one is the average number of unduplicated reporting nodes for each received packet at the sink (URN). For example, URN = 2 means that each packet received at the sink contains data from two different nodes. This metric indicates the aggregated effect. The second metric is the average end-to-end delay (ED) which is defined as the time from when an event is detected to when an associate report is received at the sink. The third is the average energy consumption for nodes within the event range (EC).

Below, we observe the effect of varying five parameters: number of nodes, aggregation ratio, event size, energy balance mechanism, and computation overhead.

A) Impact of Number of Nodes:

First, we vary the number of nodes and observe the effect on the metrics. The results are shown in Fig. 12. Obviously, in all protocols, URN, ED, and EC increase proportionally with network density. This results from the increased number of nodes within an event range for a higher network density. In Fig. 12(a), we see that the aggregation effect of both SFEB and SFEB-CL are better than that of DAA+RW, especially when network density is high. Better aggregation effect saves more energy, which explains the

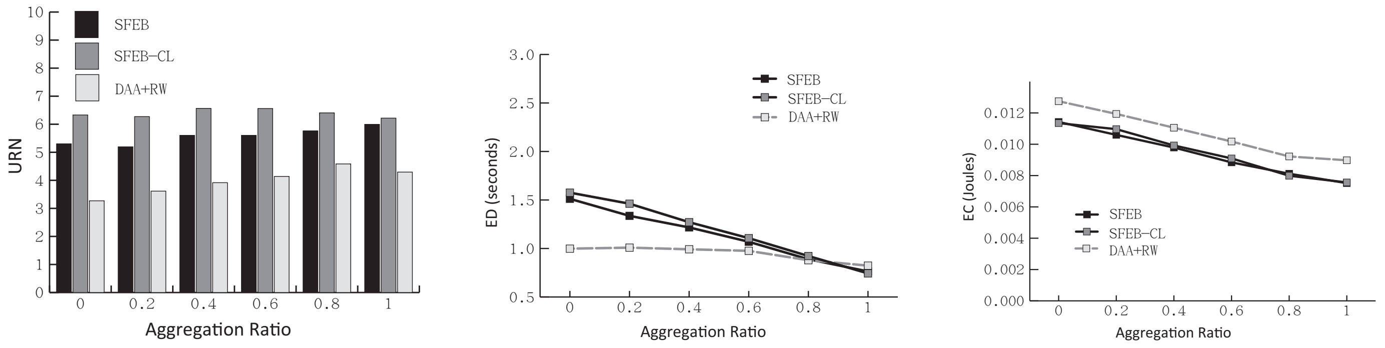


(a)

(b)

(c)

Fig. 12: Impact of number of nodes



(a)

(b)

(c)

Fig. 13: Impact of aggregation ratio

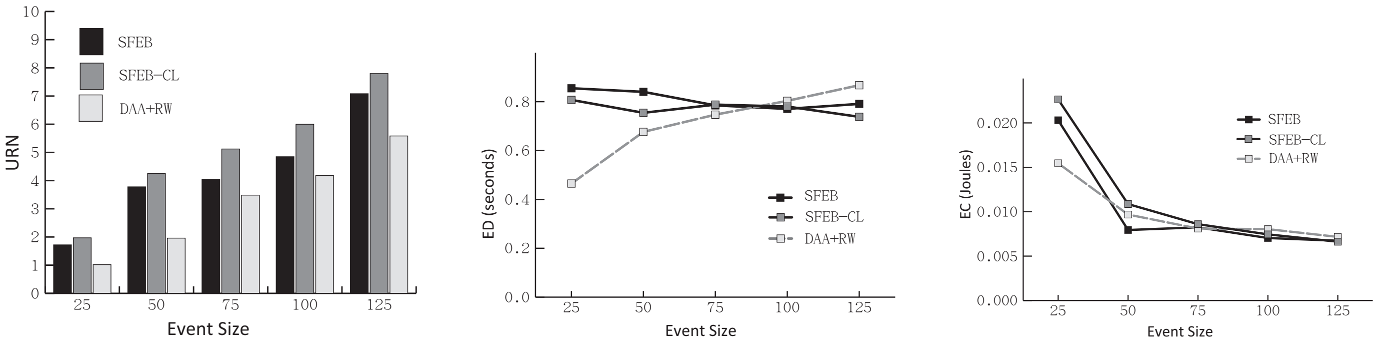
similar trends for three different protocols found in Fig. 12(c). On the other hand, as can be seen in Fig. 12(b), the average delay of our protocols is sensitive to network density changes since the length of SFEB phase one is estimated according to network density.

B) Impact of Aggregation Ratio:

In Fig. 13(a), we see the URN is insensitive to different aggregation ratios, especially for our protocols. Again, SFEB and SFEB-CL have consistently better URNs than DAA+RW. Low aggregation ratios imply longer packets, which increases end-to-end delay and energy consumption for the proposed protocols as shown in Fig. 13(b) and Fig. 13(c), respectively. To achieve early aggregation, our protocols wait longer in phase one. If aggregation ratio is low, such longer waiting will be in vain. Thus, we comment that the proposed SFEB and SFEB-CL are more suitable for an environment where aggregation ratio is not low.

C) Impact of Event Size:

In this experiment, various event sizes are tested. Increasing event size has similar effect as increasing the number of nodes within an event range. Thus, the URN comparison in Fig. 14(a) has similar trends as those in Fig. 12(a). However, increasing event size has little influence on data aggregation time in SFEB



(a)

(b)

(c)

Fig. 14: Impact of event size

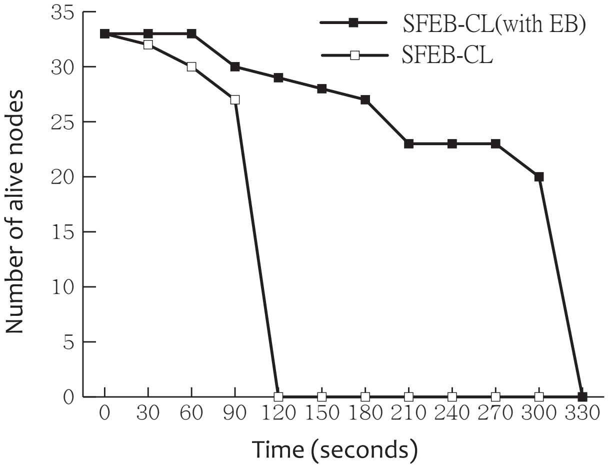
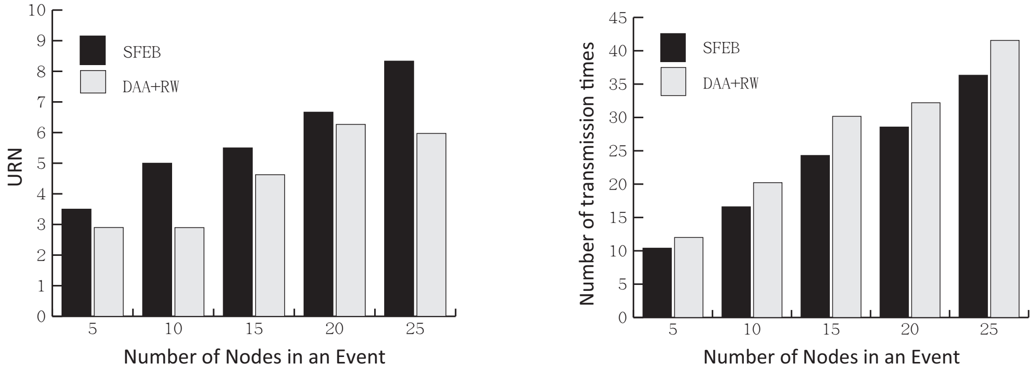


Fig. 15: Impact of energy balance mechanism

phase one, as shown in Fig. 14(b). On the other hand, DAA+RW spent much more time when event range is enlarged. Increasing event range implies more packets can be aggregated which reduces energy consumption accordingly, as shown in Fig. 14(c). When the event range is 75 or higher, all the protocols have similar power consumption. Therefore, SFEB-CL has better efficiency since the received packets at the sink for SFEB-CL is 20 to 30% higher than the other two protocols.

D) Impact of Energy Balance Mechanism:

In this experiment, we verify the effectiveness of the proposed energy balance mechanism. Here we observed the consumed energy of transmitting and receiving in SFEB phase one. Different from the others, in this simulation, a total of 600 nodes are deployed while the event radius is 75 meters. In other words, there are about 33 sensor nodes in the event range. Each sensor node has an initial energy of 0.07 Joules. The simulation is repeated until no PA can be found. We do not consider the DAA+RW protocol since no aggregator is selected in DAA+RW. The results are shown in Fig. 15. The S_p is 2 and the S_d is set



(a)

(b)

Fig. 16: Impact of real system implementation on (a) number of nodes and (b) number transmission times

to 3 seconds in the SFEB-CL with energy balance mechanism, denoted as SFEB-CL (with EB). It can be seen that the network lifetime is significantly extended by SFEB-CL (with EB) when compared with SFEB-CL. This verifies that the proposed energy balance mechanism, although it is very simple, helps prolong network lifetime.

E) Impact of Computation Overhead:

Lastly, since SFEB has higher computation overhead when compared to DAA+RW (O(c^2log$c$) to O(1)), we investigate if computation overhead is an issue in SFEB in terms of power consumption. The power consumption for computation heavily depends on the hardware in use. A reasonable setting for the ratio of power consumption to send a single bit compared to a single instruction is between 220 and 2900 for different platforms [28]. This means a single instruction consumes 3.68×10^{-9} to 2.79×10^{-10} Joules in our simulation environment. For a 200-node network, the average number of neighbors (contenders) for each node is about 6.28. Thus, the computation power consumption per node is around $(6.28)^2$log$(6.28) \times 10 \times 3.68 \times 10^{-9}$ Joules for a ten-event experiment. Adding this expense to SFEB, we find that the computation overhead is quite small. For example, when a network consists of 200 nodes, computation occupies only 0.017$\%$ of total power consumption. This shows that the computation power consumption is trivial and can be ignored.

From the results of the first three experiments, we know that SFEB-CL always outperforms SFEB in terms of URN. The delay and additional overhead of produced by SFEB-CL is not a problem because SFEB-CL has similar (sometime even lower) ED and EC when compared with SFEB. This verifies the benefit of the proposed CTS priority classifying mechanism in SFEB phase two.

VI. Real System Implementation

We implemented SFEB and DAA+RW on TinyOS 2.x with the Octopus II platform. In our implementation, 25 nodes were deployed to form a 5×5 grid network. Each node is capable of communicating directly to its vertical and horizontal neighbors. The sink node was placed at the lower right corner. The value of ρ was set to 1. In each experiment, an event is randomly generated within the network. We varied event sizes by triggering different numbers of nodes to observe the performance of SFEB and DAA+RW. The effect on URN and number of transmission times can be found in Fig. 16(a) and 16(b), respectively. Each value in the figures is the average of ten experiments. As expected, SFEB performs better than DAA+RW. Note that the enhancement for DAA+RW is not proportional to the event size. This is because DAA+RW is sensitive to the location where the event occurs. An event that is far away from the sink node may produce insignificant aggregation effect. On the contrary, for SFEB, the aggregation enhancement increases in direct proportion to event size. This proves that SFEB is a more stable data aggregation mechanism.

VII. Conclusion

In this paper, we proposed a structured-free protocol for data aggregation (SFEB). To achieve early aggregation, data aggregators are selected at the earliest possible stage. We also scheduled the data aggregation among the aggregator neighbors and suggest an energy balance mechanism. We have verified that our protocols enhance data aggregation efficiency as well as reduce energy consumption through analysis, simulation, and real system implementation. It is believed that using the proposed mechanisms can improve productivity in WSNs.

References

[1] T. van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *ACM Sensys*, 2003, pp. 171–180.

[2] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *INFOCOM*, vol. 3, 2002, pp. 1567–1576.

[3] C.-M. Chao and L.-F. Lien, “Rendezvous Enhancement in Arbitrary-Duty-Cycled Wireless Sensor Networks,” in *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2011, pp. 497–504.

[4] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, “Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[5] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks,” in *Proceedings of ACM Sensys*, 2003, pp. 28–39.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000, pp. 3005– 3014.

[7] D. Li, J. Cao, M. Liu, and Y. Zheng, “Construction of Optimal Data Aggregation Trees for Wireless Sensor Networks,” in *Proceedings of the 15th International Conference on Computer Communications and Networks*, 2006, pp. 475–480.

[8] K.-W. Fan, S. Liu, and P. Sinha, “Structure-Free Data Aggregation in Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 929–942, 2007.

[9] M. Handy, M. Haase, and D. Timmermann, “Low Energy Adaptive Clustering Hierarchy with. Deterministic Cluster-Head Selection,” in *Mobile and Wireless Communications Network*, 2002, pp. 368–372.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” *IEEE Transaction on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[11] O. Younis and S. Fahmy, “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366 – 379, 2004.

[12] H. Yang and B. Sikdar, “Optimal Cluster Head Selection in the LEACH Architecture,” in *Performance, Computing, and Communications Conference*, 2007, pp. 93–100.

[13] H. Khamfroush, R. Saadat, and S. Heshmati, “A New Tree-Based Routing Algorithm for Energy Reduction in Wireless Sensor Networks,” in *Proceedings of the 2009 International Conference on Signal Processing Systems*, 2009, pp. 116–120.

[14] R. Qiu and H. Zhao, “Energy Balance Hierarchical Data Aggregation Mechanism for Wireless Sensor Network,” in *Proceedings of the 2009 International Conference on Information Engineering*, 2009, pp. 310–313.

[15] S. Lindsey, C. Raghavendra, and K. Sivalingam, “Data Gathering Algorithms in Sensor Networks Using Energy Metrics,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.

[16] W.-H. Liao, Y. Kao, and C.-M. Fan, “Data Aggregation in Wireless Sensor Networks Using Ant Colony Algorithm,” *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 387 – 401, 2008.

[17] J.-H. Zhang, H. Peng, and T.-T. Yin, “Tree-Adapting: an Adaptive Data Aggregation Method for Wireless Sensor Networks,” in *International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–5.

[18] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-Free Localization Schemes for Large Scale Sensor Networks,” in *MobiCom*, 2003, pp. 81–95.

[19] C. Liu, K. Wu, and T. He, “Sensor Localization with Ring Overlapping Based on Comparison of Received Signal Strength Indicator,” in *Mobile Ad-Hoc and Sensor Systems*, 2004, pp. 516–518.

[20] D. Niculescu and B. Nath, “DV Based Positioning in Ad Hoc Networks,” *Journal of Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[21] V. Vivekanandan and V. Wong, “Concentric Anchor Beacon Localization Algorithm for Wireless Sensor Networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2733–2744, 2007.

[22] M.-J. Tsai, H.-Y. Yang, and W.-Q. Huang, “Axis-Based Virtual Coordinate Assignment Protocol and Delivery-Guaranteed Routing Protocol in Wireless Sensor Networks,” in *INFOCOM*, 2007, pp. 2234–2242.

[23] A. Caruso, S. Chessa, S. De, and A. Urpi, “GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks,” in *INFOCOM*, 2005, pp. 150 – 160.

[24] Y. Liu, L. Ni, and M. Li, “A Geography-Free Routing Protocol for Wireless Sensor Networks,” in *High Performance Switching and Routing*, 2005, pp. 351–355.

[25] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic Routing without Location Information,” in *MobiCom*, 2003, pp. 96–108.

[26] J. Pach and P. K. Agarwal, *Combinatorial Geometry, 3rd ed.* New York: Wiley-Interscience, 1995.

[27] C.-S. Hsu, J.-P. Sheu, and Y.-C. Tseng, "Minimize Waiting Time and Conserve Energy by Scheduling. Transmissions in IEEE 802.11-based. Ad. Hoc Networks," in *Proceedings of the 10th International Conference on Telecommunications*, 2003, pp. 393–399.

[28] J. Pach and P. K. Agarwal, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2007.