RDAG: A Structure-free Real-time Data Aggregation Protocol for Wireless Sensor Networks

Mohammad Hossein Yeganeh
Science and Research Branch, Islamic Azad University
Department of Computer Engineering
Tehran, Iran
m.yeganeh@srbiau.ac.ir

Hamed Yousefi
Sharif University of Technology
Department of Computer Engineering
Tehran, Iran
hyousefi@ce.sharif.edu

Naser Alinaghipour
Science and Research Branch, Islamic Azad University
Department of Computer Engineering
Tehran, Iran
n.alinaghipour@srbiau.ac.ir

Ali Movaghar
Sharif University of Technology
Department of Computer Engineering
Tehran, Iran
movaghar@sharif.edu

Abstract

Data aggregation is an effective technique which is introduced to save energy by reducing packet transmissions in WSNs. However, it extends the delay at the intermediate nodes, so it can complicate the handling of delay-constrained data in event-critical applications. Besides, the structure-based aggregation as the dominant data gathering approach in WSNs suffers from high maintenance overhead in dynamic scenarios for event-based applications. In this paper, to make aggregation more efficient, we design a novel structure-free Real-time Data AGgregation protocol, RDAG, using a Real-time Data-aware Routing policy and a Judiciously Waiting policy for spatial and temporal convergence of packets. Extensive simulations in NS-2 verify the superiority of RDAG in WSNs.

1 Introduction

The tendency to use high performance low cost products in wireless communications technology has led to the rapid development of Wireless Sensor Networks (WSNs) [2]. A distributed WSN is usually a self-organized system composed of a large number of small computational sensor nodes densely deployed in sorts of regions to collect various physical or environmental conditions, such as odor, vibration, pressure, light, temperature, and so on. They have a wide range of applications such as habitat and environment monitoring, military applications and real-time target tracking on battle fields, disaster and emergency surveillance, and biological hazard detection.

However, nodes have batteries with limited energy and it is difficult to supplement capacity due to the harsh environment where they are deployed. To achieve long-lived wireless multi-hop networks, it is necessary to consider how to save energy in the conception of designed algorithms and protocols for WSNs where communication costs are usually more expensive than computing costs [11].

In typical communication scenarios of WSNs, each node covering a small section of the deployment area collects data from the environment and then sends it to a sink where people may access it via the Internet. Here, it is highly possible that nodes located at the monitored area sense redundant or repeated data. Therefore, much energy is dissipated if all these data are forwarded through the network. Data aggregation is a promised technique exploiting in-network processing to conserve energy by eliminating the inherent redundancy of raw data and reducing the number of packet transmissions [11]. To aggregate data, every node combines all the received packets with its own packet into a single packet of fixed-size according to some aggregation function such as logical and/or, average, maximum, or minimum, and then forwards it to upper nodes.

Here, packets containing redundant information should be gathered together to reduce the number of transmissions. Therefore, the aggregation strategies should make packets more spatially and temporally convergent [9, 30]. A routing scheme, a spanning inward architecture of the communication topology rooted at the sink of the aggregation, is an

1533-2306/11 $26.00 © 2011 IEEE
DOI 10.1109/RTCSA.2011.70

important issue required to be considered [27]. Actually, different streams are aggregated if they happen to intersect on their way to the sink. What's more, we have to hold packets at intermediate nodes to promote aggregation efficiency. Thus, the main challenge of data aggregation is how to build up proper routes and also apply efficient timing control mechanisms to the packets in such a way that they have more chances to meet at the same node at the same time.

The present routing schemes for data aggregation rely generally on a structured architecture, such as cluster-based [15, 28] and tree-based [5, 6, 7, 8, 17, 19, 24, 25, 26]. However, in a dynamic environment, the benefit from structured data gathering may not compensate for the construction and maintenance overhead. On the contrary, the structure-free approaches achieve data aggregation using local information, so they do not spend extra energy to build a structure [3, 9].

Besides, different streams are aggregated if they happen to intersect on their way to the sink. Approaching this goal, we have to hold packets at intermediate nodes to promote aggregation efficiency. More waiting time can lead to the collection of more data, the increase of aggregation gain, and vice versa. Therefore, data aggregation has a tradeoff relationship with the delay. However, there are many real-time applications, especially in intruder tracking, medical care, fire monitoring, security surveillance, and structural health diagnosis, which need the time stringent data delivery. Here, outdated information would be irrelevant and even lead to negative effects on the system monitoring and control. Thus, a great challenge is posed to WSNs in developing data aggregation methods because they extend the queuing delay at the intermediate nodes and can consequently complicate the handling of delay-constrained data. Therefore, a key issue is how to effectively route and hold data so that real-time applications can be supported with a minimum power-consumption.

In this paper, we introduce an efficient and scalable Real-time Data AGgregation protocol, RDAG, to conserve energy by taking two mechanisms for spatial and temporal convergence of packets into account without incurring the overhead of constructing a structure while meeting the desired level of timeliness. To the best of our knowledge, this is the first study on the topic of structure-free aggregation of delay-constrained data in WSNs.

The main contribution is to maximize information fusion along the communication routes within the stipulated time bound. To make this idea work, we have to answer the following two questions: "How should data aggregator nodes be selected?" and "How much time do we need to wait for data aggregation?" To answer the first question, a Real-time Data-aware Routing policy which makes decisions on the fly for efficient spatial aggregation of data as there is no preconstructed structure is proposed. And to answer the second one, a Judiciously Waiting policy which takes advantage of the available slack by delaying packets on their way to the sink as long as their deadlines are not missed is proposed. The obtained results in the simulation environment show that RDAG presents an excellent performance in respect to aggregation gain, miss ratio, and energy efficiency.

The rest of the paper is organized as follows. Section 2 summarizes the background and related work. In Section 3 the proposed scheme and its specifications are explained. Simulation results are presented and discussed in Section 4. Finally, Section 5 concludes the paper and discusses some future directions.

2 Related work

2.1 Real-time routing

Several routing algorithms have been developed with the aim of providing timeliness in WSNs [4, 12, 14, 18, 20, 22, 23]. Here, we briefly review some of the previous works in the field. SPEED [14] implements the end-to-end transmission delay control. It finds out the neighbors' information using a beaconing mechanism and chooses the next hop based on transmission velocity and local geographical information. Moreover, it utilizes a back pressure rerouting mechanism to avoid routing traps. MMSPEED [12], an extended version of SPEED, can provide different deadlines and packet reliabilities. Moreover, R2TP [18] is a real-time routing protocol, which utilizes multipath forwarding in such a distributed way to accomplish reliable transmission in WSNs. However, MMSPEED and R2TP are similar to SPEED in that they do not consider energy expenditure in data forwarding. This issue results in quick energy exhausting of some nodes and makes the real-time characteristic and the network lifetime worse and worse. RAP [22] prioritizes real-time traffic using velocity monotonic scheduling through a differentiated MAC layer. ARP [23] considers not only the real-time requirement but also the energy index synthetically. It computes the required transmission velocity of data packets in each hop and chooses the next node according to both transmission velocity and residual node energy. RPAR [4] tries to optimize power consumption by regulating the transmission power in real-time applications. This approach is, however, affected by anomalous behavior in heavy traffic conditions, which tends to favor network congestion. Hence, RPAR increases the transmission power that worsens the situation. In THVR [20], routing decisions are made based on two-hop neighborhood velocity integrated with the residual energy awareness mechanism. However, it might lead to high computing complexity and heavy message exchange overhead to enhance the service quality of real-time packet delivery in WSNs.

2.2 Data aggregation

Data aggregation, by reducing the number of transmissions, is an effective approach to save energy. Spatial and temporal convergence during transmission are two necessary conditions for aggregation [9, 30]. To achieve a higher aggregation ratio, one category focuses on establishing a proper routing scheme and the other focuses efficient timing control. However, in data aggregation protocols, most of the present schemes rely on the static structures, such as cluster-based or tree-based.

In the cluster-based approach sensor nodes are organized into clusters. There is a cluster head in each cluster which is responsible for aggregating data from all the sensors in the cluster and transmitting the concise digest to the sink node. LEACH [15] and HEED [28] are two typical examples.

Tree-based data aggregation protocols organize sensor nodes into a tree where data aggregation is performed at intermediate nodes along the way to the sink node. One of the main aspects of tree-based networks is the construction of an energy-efficient data-aggregation tree, such as Steiner Minimum Tree (SMT) for multicast algorithms which can be used in designing data aggregation protocols [6, 24]. In energy-aware distributed heuristic (EADAT) [7], sensors with higher residual power have a higher chance to become a non-leaf tree node and thus extends the network lifetime in terms of the number of live nodes. The Power-Efficient Data gathering and Aggregation Protocol (PEDAP) [26] computes a minimum spanning tree over the WSN with transmission overhead as the link cost to minimize the total energy expended in each communication round. In cascading timeout [25], nodes schedule their timeout based on their own positions in the aggregation tree. This mechanism allocates a smaller waiting time for nodes farthest from the sink. A parent node's timeout happens after its children's timeout thus enabling a node to collect information from all its children. In [5], the authors introduce a dynamic aggregation time assignment for tree structure based on the number of children nodes of the root node. The complicated updating process of aggregation time causes the scheme to be very sensitive to little changes in the number of child nodes when the timeout happens. Literature [8] proposes a tree-based data aggregation method in real-time sensor networks. It constructs an energy efficient data aggregation tree with theoretically bounded energy cost under the latency constraint for data gathering. The data aggregation tree is constructed through the power level adjusting of sensor nodes in centralized methods. The scheme [17] proposes a heuristic algorithm for constructing data aggregation trees that minimize total energy cost under the latency bound. It develops an analytic model for IEEE Standard 802.15.4 CSMA-CA to compute the worst case delay for a sensor node in the unit of the number of time slots based on the parent node degree and the required success transmission probability. In Adaptive Time Control (ATC) [19] the locations of the nodes and the number of children in the data aggregation tree determine the aggregation timeout for a node. Thus, by ensuring sufficient time to process data from the children, it maximizes the opportunity for data aggregation.

Such structured mechanisms perform well in a stable environment when nodes function properly all the time. However, in practical environments where nodes may fail unexpectedly and also, in event-based applications, the benefit from structured gathering may not compensate for the construction and maintenance overhead. On the contrary, the structure-free approaches do not spend extra energy to build any structure. Instead, they achieve data aggregation using local information. By identifying the limitations of the static routing schemes for the data aggregation, the scheme [31] proposes Dynamic Convoy Tree-based Collaboration (DCTC) to optimize the tree reconfiguration schemes in event-based applications. However, DCTC involves heavy message exchanges. Data-Aware Anycast (DAA) [9] is the first proposed structureless data aggregation protocol that can achieve high aggregation without incurring the overhead of structure approaches. DAA uses anycast to forward packets to one-hop neighbors that have packets for aggregation and also uses Randomized Waiting (RW) at the source nodes for each packet to introduce artificial delays and increase temporal convergence. In [10] a semistructured approach utilizes DAA in a dynamic forwarding algorithm to support network scalability on an implicitly constructed structure composed of multiple shortest path trees. In SFEB [3], the structure-free and energy-balanced data aggregation protocol, the two-phase aggregation and dynamic aggregator selection enable both efficient data gathering and balanced energy consumption. In Phase One, using the concept of "gather before transmit", some data collecting nodes are selected first to gather their neighbors' sensing data as many packets as possible. Then, these aggregators send the collected packets back to the sink at Phase Two. The scheme [30] designs an effective Data Aggregation mechanism Supported by Dynamic Routing (DASDR) which can adapt itself to different scenarios without incurring much overhead. Enlightened by the concept of potential field in the discipline of physics, the dynamic routing in DASDR is designed based on two potential fields: the depth potential field which guarantees that packets will reach the sink at last and the queue potential field which makes packets more spatially convergent. Cooperating with a timing scheme similar as that in [25], this dynamic routing scheme can efficiently aggregate data.

3 RDAG

In this section, we introduce the network model and our structure-free Real-time Data AGgregation protocol (RDAG), respectively.

3.1 Network model and assumptions

We consider a set of sensor nodes distributed randomly into the two dimensional area and there exists one sink node that collects information from the sensors. Each node learns its own location and the geographic positions of the sink. The sink has no resource limitation and the sensors are battery-operated with limited energy and the same physical capabilities. Once their energy exhausts, the sensors cannot work anymore. What's more, the source nodes may transmit different types of packets and the intermediate nodes are responsible for performing in-network aggregation of same-type packets. Hence, a FIFO (First In First Out) queue is employed to hold packets for aggregation. The nodes which are not adjacent conduct data communication through hop-by-hop.

3.2 Protocol overview and properties

As mentioned above, maintaining a fixed structure for aggregating delay-constrained data is not efficient in event-based WSNs. RDAG is a distributed algorithm that provides a structure-free transmission environment for real-time data aggregation. It involves the mechanisms in such a way that the packets have more chance to meet at the same node (spatial aggregation) at the same time (temporal aggregation) while being transmitted to the sink. To achieve these objectives, our proposed protocol provides a two-fold contribution: (1) Real-time Data-aware Routing policy and (2) Judiciously Waiting policy. We describe them below.

3.2.1 Real-time Data-aware Routing policy

In this section, we present RDR, a Real-time Data-aware Routing supported by efficient data aggregation. Considering the fact that all intermediate nodes delay the received packets to aggregate more data, it is important for a node to know, at now, which next hop neighbor nodes can achieve better aggregation performance while surmounting real-time necessity. On the other hand, to determine the next hop, we must satisfy real-time and data aggregation requirements. Hence, two questions are posed. First, what is the real-time policy to implement data aggregation for delay-constraint packets. Second, are there any nodes in the radio range of the current node that have homogeneous elements, i.e., data of the same type.

In real-time applications, a packet's TTL (Time To Live) field has an important role in making routing and scheduling decisions. Thus, before node C forwards a packet, it computes the required velocity based on the progress made toward the sink node and the packet's TTL, as follows:

$$V_{req} = \frac{d(C,Sink)}{TTL} \tag{1}$$

where $d(C, Sink)$ is the Euclidean distance between current node C and the sink node. It is important to note that the deadline is met if the required velocity is met at each hop [4]. Hence, the problem of meeting end-to-end deadlines is mapped to the local problem of meeting the required velocity at each hop. This policy considers the current network conditions to adapt the packet's required velocity. If a packet is late in its way to the sink node, then its required velocity increases so that it may catch up. Conversely, its required velocity decreases if the packet is early.

We periodically measure EHD, Estimated one-Hop Delay, including channel contentions, packet transmissions, and queuing delay as in [21]. Based on the velocity requirement and the information provided for the estimated delay EHD, node N in the neighbor set is an eligible forwarding choice if it is closer to the destination and the velocity it provides; $V_{relay}(C, N)$, is equal to or greater than the packet's required velocity V_{req} [21]. Relay velocity is calculated by dividing the advance in the distance to the next hop relay node by the estimated delay to forward the packet to that node [14]:

$$V_{relay}(C, N) = \frac{d(C,Sink) - d(N,Sink)}{EHD} \tag{2}$$

where EHD, as we mentioned before, is the time it takes to forward a packet from a current node to the next hop relay node. Thus, considering that the application is delay-constrained, a node in our routing scheme can be selected as the next hop relay node provided it is an eligible forwarding choice. On the other hand, from the nodes in the neighborhood table, a node is eligible for data transmission if the real-time requirement has been satisfied.

Next hop relay nodes are prioritized based on real-time policy and aggregation efficiency. To achieve a higher aggregation ratio, RDR makes a dynamic forwarding decision by sending the packets to the node which has more same-type packets in its queue since in this way packets will be more spatially convergent. What's more, it is easy for a node to get the queue length of its neighbors [30]. Our dynamic routing scheme only uses the local information of a node to make the routing decisions, therefore it is simple and scalable.

Among the eligible forwarding nodes, node i with a higher fitness value, Fit_i, has a higher priority than the others.

$$Fit_i = Q + \delta(Q) \times \left(w \times (1 - (\frac{V_{req}}{V_{relay}(C,i)})) + (1-w) \times (\frac{E_{Rem}(i)}{E_{Init}(i)}) \right) \quad (3)$$

where Q is the number of same-type packets at node i normalized to its queue size. Moreover, $\delta(Q) = 0$ if the forwarder has some packets of the same type. Here, we will decide only based on the queue occupancy ratio and send the packets to the node having a larger Q value, so in this way packets will be more spatially convergent. On the other hand, a node which has more packets in its queue would have a higher priority to be selected as the next hop. However, if the forwarder does not have any packets of the same type, $\delta(Q) = 1$, we will decide only based on the real-time routing policy to assign the priority. Our algorithm uses factor $(1-w)$ to provide energy awareness in real-time routing as packets get closer to the sink based on the bottleneck sphere theorem [29]. This bottleneck is placed near the sink node location where all nodes have the highest energy consumption. When all eligible nodes in the bottleneck sphere fail due to the depletion of energy, the sensing data outside this sphere will not reach the sink on time, which causes quality failure. Hence, we select the weight w, formally as:

$$w = \frac{TTL}{TTL_{max}} \quad (4)$$

where TTL_{max} is the packet end-to-end deadline and TTL is used to indicate how much time remains for the packet to arrive at the sink. As packets get closer to the sink, the value of $(1 - w)$ increases, so the effect of residual node energy is more highlighted in routing decisions where $E_{Rem}(i)$ and $E_{Init}(i)$ are the residual and initial energy of eligible node i. Using this factor enables even balance of traffic between the eligible nodes along the path to the sink node, and especially in the bottleneck sphere.

At last, a node will be picked out of the eligible neighbor nodes while its priority is the highest. However, if there is no eligible node in the neighbor set, the back pressure rerouting mechanism is aimed instead of packet dropping [14].

Using the RDR approach can result in aggregation as early as possible on the routes to the sink. Moreover, RDR is tolerant to interference and node failures due to the dynamic routing in intermediate nodes; so it is very robust even in unreliable WSNs. Finally, packets will converge to the best aggregation points dynamically without explicit construction and maintenance of an aggregation structure while considering real-timeliness necessity for efficient delay-constraint data forwarding. However, in the RDR approach, packets may not be aggregated if they are spatially separated, so we use the timing control policy for temporal convergence to improve aggregation efficiency as the intermediate nodes delay the packets.

3.2.2 Judiciously Waiting policy

There is a trade-off between energy and delay because aggregation requires that some data be delayed at intermediate nodes while waiting for more data packets to be received. Hence, a key issue in the context of real-time monitoring is the calculation of waiting timeout for each forwarding packet in such a way that it can be delivered to the sink within a stipulated time bound.

Aggregation scheduling is mostly classified into three different categories in structure-based approaches as presented below [25]:

- Simple periodic: in this method each node sends out the aggregated packet to the next hop in a pre-defined period of time.
- Periodic per-hop: in this scheme each node performs aggregation as soon as it hears from all its children and then sends out one single packet.
- Periodic per-hop adjusted: this method is the same as the periodic per-hop policy, but in each node, holding time is calculated based on the position of the node in the tree structure.

However, in structure-free approaches, exploiting an unalterable timeout policy such as the periodic scheme is not applicable. Particularly, this problem is highlighted even more in data aggregation applications with limited data packet delivery time because passing the packet through different paths can result in the reception of packets with different deadlines, thus requiring different timing policy. Therefore, the main question is how long a packet could be delayed at intermediate nodes so that the aggregation gain and on-time end-to-end delivery ratio will be maximized.

Here, we propose the Judiciously Waiting policy for each packet along the way to the sink node to introduce artificial delays and increase temporal convergence for effective aggregation while meeting the time constraint of the data. We allocate the available packet's slack time, i.e., $TTL - EED$, proportionately to the remaining hop count to the sink node along the forwarding path to judiciously hold packets at intermediate nodes while surmounting soft real-time. The estimated End-to-End Delay (EED) is the time that takes to deliver the packet from current forwarding node to the sink node. To estimate EED, we periodically measure EHD, Estimated one-Hop Delay. Besides, the overall effectiveness of aggregation is dependent on when and where it actually occurs. Based on [13], aggregating data close to the source nodes is the most efficient communication for perfect aggregation functions. Therefore, in our algorithm, the waiting timeout WT for a packet at an intermediate node R_h hops away from the sink is calculated as follows:

$$WT = \frac{TTL - EED}{1 + \left(\frac{R_h - 1}{R_h}\right)}.\alpha = \frac{TTL - (R_h \times EHD)}{2 - \left(\frac{1}{R_h}\right)}.\alpha \qquad (5)$$

where α is a constant factor used to leave some remaining time as a safety margin to ensure that the deadline would be met. Moreover, the remaining hop count to the sink node is formally calculated by $R_h = \frac{d(C,Sink)}{d(C,N)}$ where $d(C, Sink)$ and $d(C, N)$ are the distance from the current node to the sink and to the next hop forwarding node (see Section 3.2.1), respectively. Based on (5), by decreasing the remaining hop count a lower part of slack time is used for data aggregation as the packet moves closer to the sink. Finally, the packet uses its entire remaining deadline as a slack time in the current node if the next hop node is the sink node (i.e., $R_h = 1$).

Thus, taking advantage of the available slack if any, our waiting time policy not only improves aggregation efficiency by judiciously delaying packets at intermediate nodes but also tolerates transient periods of high contentions without requiring any synchronization among sensor nodes.

4 Performance evaluation

In this section, we evaluate the performance of our algorithm via simulation. We conducted our simulations using NS-2 [1]. The goal of the simulation is to show that our proposed protocol, RDAG, can outperform other important real-time routing and dynamic data aggregation protocols using two mechanisms for spatial and temporal convergence of delay-constrained packets - Real-time Data-aware Routing policy and Judiciously Waiting policy. The results are compared with SPEED [14], the most well-known real-time routing protocol, to show aggregation efficiency of RDAG and also DASDR [30], a recent dynamic protocol for efficient data aggregation, to show the efficiency of RDAG for real-time applications in WSNs. DASDR designs a dynamic routing scheme based on the depth-queue potential field which makes packets more spatially convergent while cooperating with cascading timeout policy for temporal data convergence.

The simulation parameters for our model are mentioned in Table I. The communication parameters are mostly chosen in reference to the Berkeley mote specifications [16]. We also assume a simple and realistic energy model using a distance based formula [15]. Let E_{sense} be the energy depleted for sensing. Moreover, E_{agg} is the energy employed for data aggregation in the case when the node acts as an aggregator. In the transmitting mode, energy is spent in the electronic components, E_{elec}, as well as in the front-end amplifier, E_{amp}, which supplies the power for the actual RF transmission. In the receiving mode, energy is consumed entirely by the transceiver electronics, E_{elec}.

Table 1. Simulation Parameters

Terrain	$200m \times 200m$
Node Number	100
Topology	*Grid*
Initial Node Energy	$1\ J$
Bandwidth	$200\ Kb/s$
Radio Range	$40m$
Propagation Model	*Two Ray*
Payload Size	$50\ Byte$
E_{elec}	$50\ nJ/bit$
E_{sense}	$0.083\ J/s$
E_{amp}	$10\ pJ/bit/m^2$
E_{agg}	$5\ nJ/bit/signal$

We ran the simulation with several parameters, including data rate and aggregation limit, where 6 nodes randomly chosen from the left side of the terrain, send periodic data to the sink placed at the middle of the right side of the terrain. To create a dynamic event-based environment, each source node assigns a random type to its periodically generated data packet. We define 3 types of data and only same-type packets could be aggregated at intermediate nodes along the way to the sink node. The simulation ends as soon as 5000 packets are received at the sink. Therefore, we add a field to each aggregated packet to specify how many effective pieces of information are contained in this packet. We derived different aggregation limits D (4, 8, ∞) from different values of aggregation ratio (0, 0.5, 1) in [9]. D actually represents the maximum number of packets whose information can be aggregated by a aggregator node into one packet. The packets wait at the queue until the number of accumulated packets is equal to the maximum aggregation limit D. Moreover, according to Judiciously Waiting policy in each node, different TTLs of the incoming packets make their waiting time different. Thus, the aggregation timer is continuously readjusted to the minimum waiting time of the incoming packets, if necessary, to meet their deadlines. Finally, the packets accumulated in the aggregation queue are flushed when the queue is full or the timer expires. After expiration, the aggregated result has a TTL equal to the minimum TTL of the packets and then is sent to the next hop forwarding node. We set α=0.7, as in [21], and increase the packet generation rate step by step from 1 to 100 pps (packets/second) and then compare RDAG with the existing protocols in the terms of aggregation gain, miss ratio, and average energy consumption where:

- Aggregation gain is defined as the measure of reduction in the communication traffic due to the aggregation. Thus, it is the ratio of traffic reduction due to

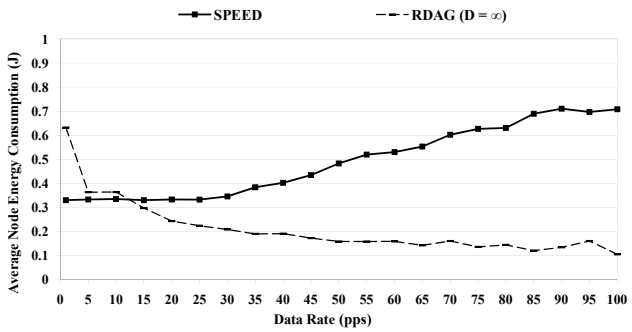

Figure 1. Average node energy consumption vs. Data rate

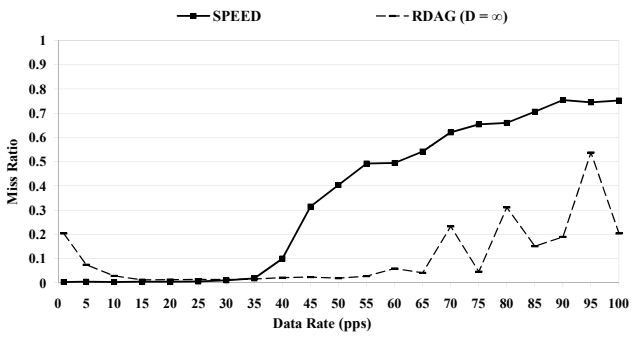

Figure 2. Miss ratio vs. Data rate

aggregation to the total traffic without aggregation [5]

- Miss ratio, the most important metric in soft real-time systems, is defined as the percentage of packets that do not meet their end-to-end deadlines.

From Figures 1 and 2 it can be seen that RDAG outperforms SPEED in terms of energy consumption and miss ratio, especially in higher data rates, where the aggregation queue is unlimited and generated packets have the end-to-end deadline of 500 ms. In SPEED no packet gets aggregated along the way to the sink; thus, more packets are injected into the network. However, RDAG uses a Judiciously Waiting policy to take the advantage of available slack for efficient aggregation, so it conserves energy through eliminating the inherent redundancy of raw data. As the traffic increases, miss ratio increases as well because more contention can result in longer queueing delays and congestion at intermediate nodes. A higher number of missed packets can lead to a higher number of retransmissions as well as generated packets in the source nodes, and finally, a higher energy consumption in heavy traffic loads. The fluctuation in the curves of our proposed method is justifiable considering that RDAG selects different routes in different traffic loads for data forwarding in a real-time aggregation-aware scheme, so there is a kind of randomness in routing.

Figure 3 compares the aggregation ability of different dynamic aggregation protocols and actually depict the efficiency of waiting time policy in terms of aggregation gain for various aggregation limits under different traffic loads when the end-to-end deadline is 700 ms. As expected, when the data rate increases, more packets have the chance to meet each other, so the aggregation ability increases. Moreover, our proposed algorithm outperforms DASDR in terms of aggregation gain since in our method, each aggregator node has an accurate estimation of delays along the ways to the sink node, hence it could collect more data from its neighbors in a judicious manner. However, in DASDR the scheduling is based on cascading timeout policy which can not appropriately delay packets at intermediate nodes, thus it has less aggregation gain. In addition, the effect of aggregation limit on aggregation gain is shown in Figure 3. By increasing the aggregation limit in our protocol more number of same-type packets can be aggregated into one packet, so the aggregation gain increases.

Figure 4 shows the packet miss ratio for various aggregation limits under different data rates. As it can be seen, by increasing the data rate, the miss ratio increases as well for all different schemes. However, RDAG outperforms DASDR by combining both real-time routing and timing control for delay-constrained data aggregation. Moreover, our scheme reduces the number of packets injected into the network at intermediate nodes from sources to the sink by providing a higher data aggregation at the nodes closer to the source nodes. Thus, RDAG can handle the number of transmissions and the network traffic more effectively as the data rate increases.

Moreover, as Figure 5 shows, RDAG has a higher energy efficiency than DASDR due to using an efficient real-time data aggregation policy which decreases the miss ratio, and increases the aggregation gain as well as the network lifetime.

Thus, considering the delay constraint, RDAG uses a Real-time Data-aware Routing policy for spatial convergence and a Judiciously Waiting policy for temporal convergence to increase the aggregation gain and energy efficiency and decrease the miss ratio as much as possible.

5 Conclusion

Data aggregation plays an important role in energy-constrained WSNs. This paper proposed a structure-free data aggregation mechanism, RDAG, for collecting delay-constrained data in WSNs. To Achieve this, it combines a dynamic Real-time Data-aware Routing policy and a Judiciously Waiting policy. The first policy considers data types as well as real-time decisions to select next hop neighbor

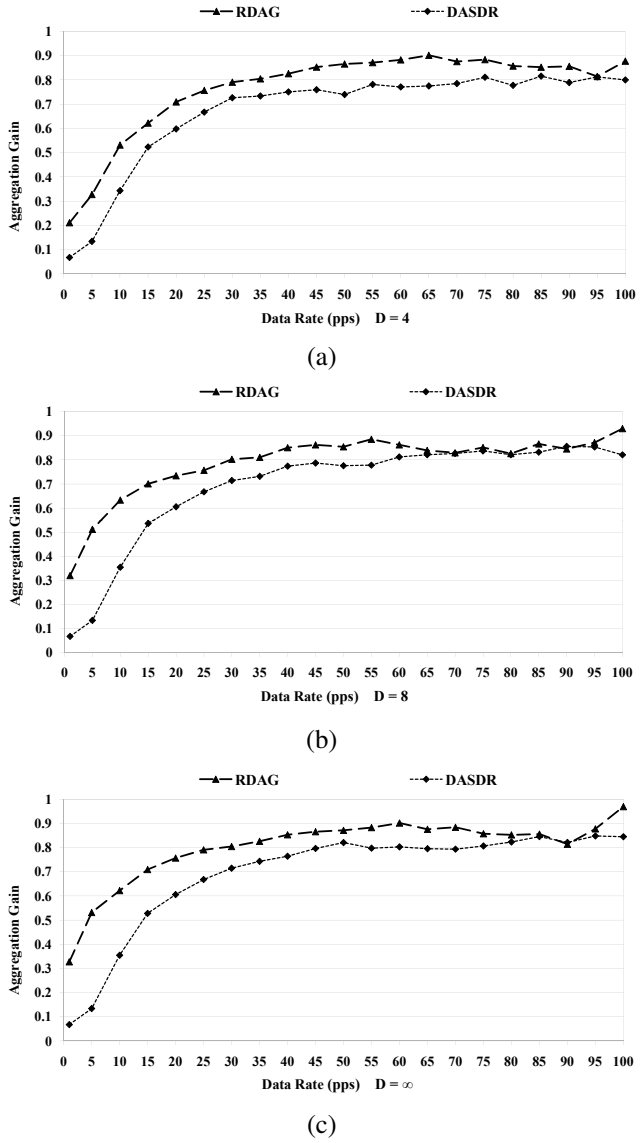

Figure 3. Aggregation gain vs. Data rate (a) $D = 4$**, (b)** $D = 8$**, (c)** $D = \infty$**.**

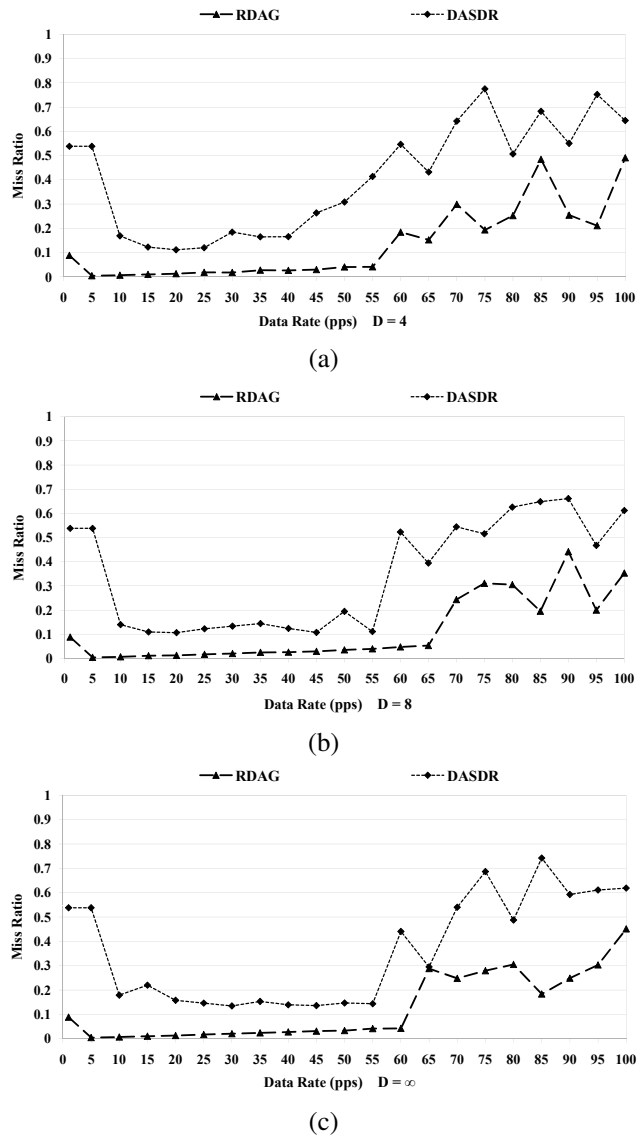

Figure 4. Miss ratio vs. Data rate (a) $D = 4$**, (b)** $D = 8$**, (c)** $D = \infty$**.**

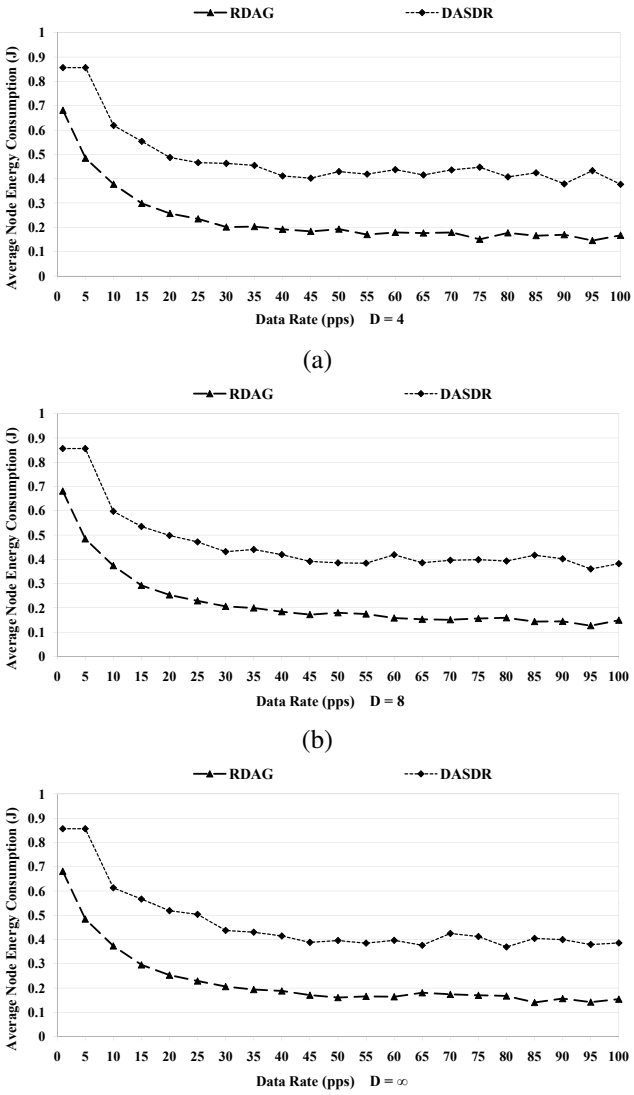

Figure 5. Average node energy consumption vs. Data rate (a) $D = 4$**, (b)** $D = 8$**, (c)** $D = \infty$**.**

nodes with better aggregation performance and increases spatial convergence during transmissions without explicit maintenance of a structure while the second one introduces artificial delays and increases temporal convergence. The results evaluated in simulation verified the superiority of RDAG in terms of aggregation gain, miss ratio, and energy efficiency. Therefore, our proposed method is very suitable for conserving energy in real-time WSNs.

The algorithm could be modified to take into account some aspects that have not been addressed in this work. These aspects could be interesting topics for future research. For instance, studying an aggregation-aware real-time routing protocol in mobile WSNs could be considered in future studies. Latency could also be evaluated during the process of data aggregation. Moreover, we plan to use other selection route algorithms to improve our routing structure.

References

[1] The network simulator, http://www.isi.edu/nsnam/ns.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

[3] C. M. Chao and T. Y. Hsiao. Design of structure-free and energy-balanced data aggregation in wireless sensor networks. In *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC'09)*, pages 222–229, 2009.

[4] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. In *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS'06)*, pages 83–92, 2006.

[5] J. Y. Choi, J. W. Lee, K. Lee, S. Choi, W. H. Kwon, and H. S. Park. Aggregation time control algorithm for time constrained data delivery in wireless sensor networks. In *Proceedings of the 63rd IEEE Vehicular Technology Conference*, pages 563–567, May 2006.

[6] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On network correlated data gathering. In *Proceedings of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, pages 2571–2582, 2004.

[7] M. Ding, X. Cheng, and G. Xue. Aggregation tree construction in sensor networks. In *Proceedings of the 58th IEEE Vehicular Technology Conference*, pages 2168–2172, 2003.

[8] H. Du, X. Hu, and X. Jia. Energy efficient routing and scheduling for real-time data aggregation in wsns. *Computer Communications*, 29(17):3527–3535, 2006.

[9] K. W. Fan, S. Liu, and P. Sinha. Structure-free data aggregation in sensor networks. *IEEE Transactions on Mobile Computing*, 6(8):929–942, 2007.

[10] K. W. Fan, S. Liu, and P. Sinha. Dynamic forwarding over tree-on-dag for scalable data aggregation in sensor networks. *IEEE Transactions on Mobile Computing*, 7(10):1271–1284, 2008.

[11] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: A survey. *IEEE Wireless Communications*, 14(2):70–87, 2007.

[12] E. Felemban, C. Lee, and E. Ekici. MMSPEED: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(6):738–754, 2006.

[13] A. F. Harris, R. Kravets, and I. Gupta. Building trees based on aggregation efficiency in sensor networks. *Ad Hoc Networks*, 5(8):1317–1328, 2007.

[14] T. He, J. A. Stankovic, L. Chenyang, and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pages 46–55, May 2003.

[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.

[16] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *Proceedings of (ASPLOS'00)*, pages 93–104, 2000.

[17] Y. Hu, N. Yu, and X. Jia. Energy efficient real-time data aggregation in wireless sensor networks. In *Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC'06)*, pages 803–808, 2006.

[18] K. Kim, S. Park, H. Park, and Y. H. Ham. Reliable and real-time data dissemination in wireless sensor networks. In *Proceedings of Military Communications Conference (MILCOM'08)*, pages 1–5, 2008.

[19] H. Li, H. Yu, B. Yang, and A. Liu. Timing control for delay-constrained data aggregation in wireless sensor networks. *Communication Systems*, 20(7):875–887, 2007.

[20] Y. Li, C. S. Chen, Y. Song, Z. Wang, and Y. Sun. Enhancing real-time delivery in wireless sensor networks with two-hop information. *IEEE Transactions on Industrial Informatics*, 5(2):113–122, May 2009.

[21] K. Liu, N. Abu-Ghazaleh, and K. Kang. JiTS: Just-in-time scheduling for real-time sensor data dissemination. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'06)*, 2006.

[22] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium*, 2002.

[23] H. Peng, Z. Xi, L. Ying, C. Xun, and G. Chuanshan. An adaptive real-time routing scheme for wireless sensor networks. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, pages 918–922, May 2007.

[24] H. F. Salama, D. S. Reeves, and Y. Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *IEEE Journal on Selected Areas in Communications*, 15(3):332–345, 1997.

[25] I. Solis and K. Obraczka. The impact of timing in data aggregation for sensor networks. In *Proceedings of the IEEE International Conference on Communications*, pages 3640–3645, 2004.

[26] H. O. Tan and I. Korpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *SIGMOD Record*, 32(4):66–71, 2003.

[27] P. Wan, S. C. Huang, L. Wang, Z. Wan, and X. Jia. Minimum-latency aggregation scheduling in multihop wireless networks. In *Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'09)*, pages 185–194, 2009.

[28] O. Younis and S. Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.

[29] P. Zeng and H. Yu. An ant-based routing algorithm to achieve the lifetime bound for target tracking sensor networks. In *Proceedings of the INFOCOM'06*, 2006.

[30] J. Zhang, Q. Wu, F. Ren, T. He, and C. Lin. Effective data aggregation supported by dynamic routing in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC'10)*, pages 1–6, May 2010.

[31] W. Zhang and G. Cao. DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Transactions on Wireless Communications*, 3(5):1689–1701, 2004.