Structure-free Data Aggregation in Sensor Networks

Kai-Wei Fan, Sha Liu, and Prasun Sinha
Department of Computer Science and Engineering
The Ohio State University
Email: {fank, liusha, prasun}@cse.ohio-state.edu

Abstract

Data aggregation protocols can reduce the cost of communication, thereby extending the lifetime of sensor networks. Prior work on data aggregation protocols has focused on tree-based or cluster-based structured approaches. Although structured approaches are suited for data gathering applications, they incur high maintenance overhead in dynamic scenarios for event-based applications. The goal of our work is to design techniques and protocols that lead to efficient data aggregation without explicit maintenance of a structure. As packets need to converge spatially and temporally for data aggregation, we propose two corresponding mechanisms - *Data-Aware Anycast* at the MAC layer and *Randomized Waiting* at the application layer. We model the performance of the combined protocol that uses both the approaches and show that our analysis matches with the simulations. Using extensive simulations and experiments on a testbed with implementation in TinyOS, we study the performance and potential of structure-free data aggregation.

Index Terms

Sensor Networks, Data Aggregation, Anycasting, Structure-free

I. Introduction

In sensor networks, the communication cost is often several orders of magnitude higher than the computation cost. For optimizing the communication cost, in-network data aggregation is considered an effective technique. The inherent redundancy in raw data collected from the sensors can often be eliminated by in-network data aggregation. In addition, such operations are also useful for extracting application specific information from the raw data. To conserve energy for a longer network lifetime, it is critical for the network to support high incidence of in-network data aggregation.

Optimal aggregation can be defined in terms of the total energy consumption for transporting the collected information from the sensors to the sink. Based on the topology of the network, location of nodes collecting the raw data, and the aggregation function, an optimal aggregation structure can be constructed. Various centralized structured approaches [1]–[4] have been proposed for aggregation in data gathering applications where all nodes periodically report to the sink. Due to the unchanging traffic pattern, structured aggregation techniques incur low maintenance overhead and are therefore suited for such applications. Various distributed structured approaches have been proposed for event-based applications [5]–[7]. However there are several limitations of structured aggregation techniques for event-based applications. First, for dynamic scenarios the overhead of construction and maintenance of the structure may outweigh the benefits of data aggregation. Second, some distributed approaches such as [5] assume that there is a well defined center-of-event and that the measured strength of the sensed signal is an indicator of the distance to the center of the event. For amorphous events, such as biological hazard, chemical hazard, or fire detection, absence of an explicit center or any evident point for optimal aggregation makes such approaches inapplicable. Third, structured approaches that centrally compute the aggregation tree [4] are not practical for dynamic scenarios due to excessive communication overhead for centralized computation. And fourth, the performance of structured approaches is sensitive to the waiting period (for data from all upstream nodes) at the intermediate nodes. A small period can lead to fewer number of aggregations and a long period can lead to higher latency. Moreover, computing the optimal period requires knowledge of the relative position of the node with respect to its entire subtree, which may not be known accurately especially for dynamic scenarios.

In [8] we explore the potential of structure-free data aggregation for event-based sensor networks. In this paper, we further reduce the number of transmissions and conduct more thorough simulations to explore the benefit of our structure-free approach. *The goal of our work is to design techniques and protocols that lead to efficient data aggregation without explicit maintenance of a structure.* To the best of our knowledge this is the first study on the topic of structure-free data aggregation in sensor networks. A combination of a partial structured approach with our structure-free approach may further improve performance, but it is outside the scope of this paper.

There are two main challenges in performing structure-free data aggregation. First, as there is no pre-constructed structure, routing decisions for efficient aggregation of packets need to be made on-the-fly. Second, as nodes do not explicitly know their upstream nodes, they can not explicitly wait on data from any particular node before forwarding their own data.

The contributions of this paper are:

- We observe that packets need to be aggregated early on their route to the sink for efficiency. Based on this observation we propose and model a MAC layer protocol for spatial convergence called Data-Aware Anycast (DAA).
- We observe that if some nodes wait for other nodes to send data, it can lead to efficient aggregation. We study the impact of Randomized Waiting (RW) for improved data aggregation.
- We perform detailed evaluation using simulations to establish the benefits of structure-free data aggregation, and observe that structure approach may not always be the best strategy for data aggregation.
- We implement the randomized waiting approach on TinyOS and test the randomized waiting approach in an indoor network of 6 Mica2 motes.

The organization of the rest of the paper is as follows. Section II presents background and related work. Section III presents the DAA and RW protocols. The performance of the combined approach that uses both DAA and RW is modeled in Section IV. The performance evaluation of the protocols using simulations is presented in Section V. Results from the indoor testbed confirming the benefits of the RW approach are presented in Section VI. Finally Section VII concludes the paper.

II. Related Work

Because of the energy constraint and relatively expensive communication cost, the computation cost of sensor nodes becomes less significant in wireless sensor networks. Pottie and Kaiser [9] report that the energy consumption for executing 3000 instructions is equivalent to sending a bit 100 meters by radio. Therefore, data aggregation, data-centric routing and in-network processing are very important to extend the lifetime of wireless sensor networks. Various protocols have been proposed to facilitate data aggregation. They can be categorized into two families: cluster-based and tree-based. This section briefly reviews these protocols by their structure categories and provides the motivation for our work.

A. Cluster-Based Approaches

LEACH [1], [2] and PEGASIS [3], [10], [11] are representatives of this family. In [1], the authors propose the LEACH protocol to cluster sensor nodes and let the cluster-heads aggregate data and communicate directly with the base station. To distribute energy consumption evenly among all nodes, the cluster-heads are randomly elected in each round. In [2], authors propose a modified version named LEACH-C, which uses the base-station to broadcast the cluster-head assignment, thus further spreading out the cluster heads evenly throughout the network and extending the network lifetime. Based on LEACH, [12] refines the cluster-head election algorithm by letting every node broadcast and count neighbors at each setup stage, where qualified potential nodes bid for the cluster head position.

Reducing the number of cluster-heads is critical to conserve energy as these nodes stay awake and transmit to the base-station using high power. Lindsey et al. designed PEGASIS [3], which organizes all nodes in a chain and lets them be the head in turn. Since there is only one head node in PEGASIS and there is no simultaneous transmissions, latency is an issue in PEGASIS. To address this, Lindsey et al. propose two chain-based PEGASIS enhancements in [10], [11]. [10] is a binary hierarchical approach for CDMA-capable sensor nodes, and [11] is a chain-based three level approach that allows simultaneous transmission for non-CDMA-capable sensor nodes. These two approaches usually save less energy than PEGASIS, but outperform PEGASIS in $Energy \times Delay$ metric. Based on both LEACH and PEGASIS, Culpepper et al. propose Hybrid Indirect Transmission (HIT) [13], a hybrid scheme of these two. HIT still uses LEACH-like clusters, but allows multi-hop routes between cluster-heads and non-head nodes.

LEACH and PEGASIS based protocols assume that the base-station can be reached by any node in only one hop, which limits the size of the network for which such protocols are applicable. The combination of CSMA, TDMA and CDMA also makes the design complex and cost-inefficient. In addition, in scenarios where the data can not be perfectly aggregated, LEACH and PEGASIS-based protocols do not necessarily have significant advantage since the cluster-head has to send many packets to the base station using high transmission power.

B. Tree-Based Approaches

Protocols in this family are built on traditional shortest path routing tree. The research is focused on how to choose a good routing metric based on data attributes to facilitate data aggregation. Directed Diffusion [14], [15] is one of the earliest to propose attribute-based routing. Data can be opportunistically aggregated when they meet at any intermediate node. Based on Directed Diffusion, the authors propose Greedy Incremental Tree (GIT) in [7], [16]. GIT establishes an energy-efficient path and attaches other sources greedily onto the established path. If more than one paths are reinforced, paths with lower energy-efficiency are pruned and related sources are reattached greedily to the unpruned path. Heidemann et al. [17] further study the effect of data aggregation on Directed Diffusion through experimentation. Krishnamachari et al. [18] compare three data-centric routing schemes, Center at Nearest Source (CNS), Shortest Path Tree (SPT) and another version of Greedy Incremental Tree (GIT) which establish the route between the sink the the nearest source first, to illustrate the advantage of data aggregation for saving energy. They observe that GIT performs the best in terms of average number of transmissions. In [19], [20] Madden et al. study the data aggregation issue in implementing a real system and propose the Tiny AGgregation Service (TAG) framework. TAG uses shortest path tree, and proposes improvements like snooping-based and hypothesis testing based optimizations, dynamic parent switching, and using child cache to obtain lost data. TAG lets parents notify their children about the waiting time to gather all data from children before transmitting, and the sleeping schedule can be adjusted accordingly. Through real field experiment, TAG shows the benefit of data aggregation, not only on saved energy, but also on data quality improvement due to less contention. Ding et al. use shortest path tree with parent energy-awareness in [21], where the neighbor node of shortest distance to the sink with higher residual energy is chosen as the parent. All the above tree-based data aggregation routing protocols need a lot of message exchange to construct and maintain the tree.

Most of these tree-based aggregation routing protocols are not designed for event tracking applications. GIT can be used in such a scenario, but it suffers from the overhead of pruning branches, which might lead to high cost in moving event scenarios. Zhang and Cao propose Dynamic Convoy Tree-Based Collaboration (DCTC) in [5]. They propose a conservative scheme and a prediction-based scheme to wake up and prune nodes from the convoy tree, and favor the latter one given reasonable prediction accuracy. They also propose the message-intensive sequential reconfiguration scheme, which is suitable for sparse networks, and heuristic-based localized reconfiguration, which is suitable for dense networks. In [6], Zhang and Cao further optimize the tree reconfiguration schemes. They compare optimized complete reconfiguration (OCR) and optimized interception-based reconfiguration (OIR), and state the OIR is suitable for small data size and small monitoring regions, while OCR is suitable for other cases. DCTC assumes the knowledge of distance to the center of the event at sensor nodes, which may not be feasible to compute with the sensed information in all tracking applications. In addition, DCTC involves heavy message exchanges, which is not desired when the data rate is high.

The other class of data aggregation protocols takes sensing information entropy into the routing metric. In [22], [23], Scaglione and Servetto propose a broadcast routing protocol to disseminate the information from each node to all other nodes in the network under the capacity constrained networks, but the broadcast scenario is different from ours. In [24], [25], Cristescu et al. first study the distributed source coding and apply its approximation to sensor networks. But source coding or even its approximation is hard to deploy since the real sent information distribution is hard to know. Authors then study explicit communication taking joint entropies among nodes into routing metric. The authors prove that to find the optimal routing tree using the new routing metric is NP-complete, and propose approximation algorithms such as Leaves Deletion approximation and Balanced SPT/TSP tree. But these algorithms are centralized. In addition, they assume the global knowledge of the information entropy of each sensor nodes and the joint entropy of each pair, which makes such approaches not practical. Pattern et al. study the impact of spatial correlation on routing in [26], and derive the optimal cluster size for these cases. However, their study are confined to some special cases.

III. Spatial and Time Convergent Protocol

For optimal aggregation, nodes must transmit their packets in a certain order. Structured approaches are designed to follow such orderings for achieving optimal aggregation. For example the transmissions should proceed from the leaves to the root for a tree. Structured approaches though suited for data gathering applications, have high overhead for event-based applications as discussed in Section I. The other extreme is to use opportunistic aggregation where packets are aggregated only if they happen to meet at the same node at the same time while being forwarded to the sink. In opportunistic aggregation, there is no overhead of aggregation structure construction, however it may result in inefficient data aggregation. Without explicit coordination, the performance of the opportunistic aggregation technique is non-deterministic, and the chance of aggregation may be limited. To avoid the overhead of structured approaches and the limitations of opportunistic aggregation, we study the design of structure-free techniques for data aggregation.

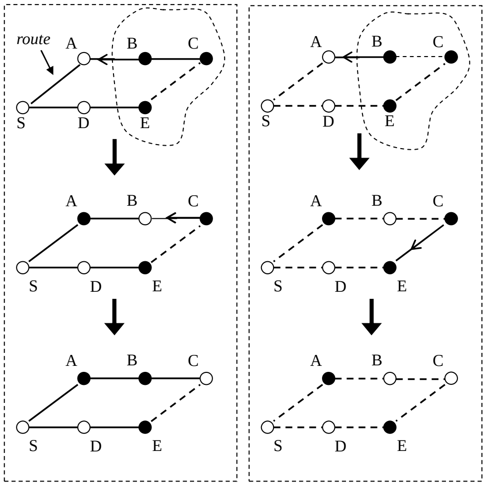
Spatial convergence and temporal convergence during transmission are two necessary conditions for aggregation. Packets have to be transmitted to the same node at the same time so they can be aggregated. Structured approaches achieve these two conditions by letting nodes transmit packets to their parents in the aggregation tree and parents wait for packets from all its children before transmitting their packets. Without explicit message exchange in structure-free aggregation, nodes do not know where they should send packets to and how long they should wait for aggregation. Therefore improving spatial convergence or temporal convergence can improve the chance of aggregation. We propose the Data-Aware Anycast (DAA) protocol for improving spatial convergence and the Randomized Waiting (RW) technique for improving temporal convergence. These two approaches are described in the rest of this section.

For the design of the structure-free convergence protocol we have the following goals.

1) **Early aggregation:** Packets must get aggregated as early as possible on their journey to the sink.
2) **Tolerance to event dynamics:** If the event's region of influence is changing due to mobility, the overhead must not increase and the aggregation performance must remain unchanged.
3) **Robust to interference:** Intermittent link failures should not affect the aggregation performance.
4) **Fault tolerance:** The aggregation performance must not be affected by node failures which are common in sensor networks.

A. Spatial Convergence

In this section we present the Data-Aware Anycast (DAA) protocol which achieves the goals described above. The idea behinds DAA is, instead of constructing an structure in advance for optimal aggregation which is impossible without global knowledge of the network and traffic pattern, an independent set among nodes which generate packets is created. Nodes in



(a) Three packets left in the network.

(b) Two packets left in the network.

Fig. 1. **Enhancing Opportunistic Aggregation with Spatial Convergence:** (a) In opportunistic aggregation, nodes send their packet along the route constructed by the routing protocol. (b) Improving spatial convergence by allowing nodes to send packets to nodes that still have packets for aggregation. S is the sink, solid lines are routes constructed by the routing protocol, dotted lines are other wireless links, and the arrows on the links represent packet transmissions. A node with data are represented with a dark circle.

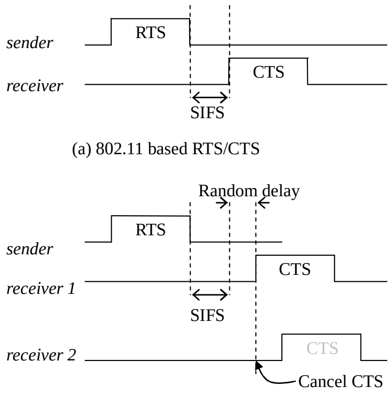
the independent set act as aggregation points such that the aggregation from nodes to these aggregation points is optimal. The independent set is created distributedly and automatically while packets are forwarded to the sink, therefore it reduces the maintenance overhead of structured approaches. The following paragraphs described the intuition and approach of DAA in detail.

When nodes send packets to the sink, they may follow different routes dictated by the routing protocol even if they are close to each other. Fig. 1 shows an example comparing opportunistic aggregation with optimal forwarding strategy. Fig. 1(a) is the packet transmissions assuming opportunistic aggregation. Fig. 1(b) shows how information about existence of data in neighboring nodes can be exploited to make dynamic forwarding decisions to achieve higher aggregation. The black nodes are nodes that have packets to send. Because there is no message exchange to construct a structure for aggregation, packets from C and E follow two different routes constructed by the routing protocol in (a). The distributed MAC protocol determines the order of transmissions in opportunistic aggregation which does not achieve any aggregation in this case. However, as shown in Fig. 1(b), if node C knows that node B does not have any packet for aggregation but node E does, it can send the packet to E for immediate aggregation.

As a result there are only two packets left in the network (as opposed to 3 for opportunistic aggregation). This process can be repeated until a node does not have neighbors with packets for aggregation, such as E in Fig. 1(b), and we call E as the aggregation point. This shows that if the routing protocol provides the freedom to the MAC layer to decide among a set of nodes (rather than a single next-hop), and if it can determine which node has packets for aggregation, efficient spatial convergence can be achieved. In typical deployment of sensor networks, nodes have multiple choices for the next-hop node. For example, in the ExScal [27] demonstration of the world's largest sensor network, each sensor has anywhere between 3 to 32 nodes in its communication range.

We present the mechanisms of the DAA approach by discussing the base approach and enhancements to the base approach.

- **DAA - The base approach:** Our approach is based on *anycasting* [28]–[30] at the MAC layer for determining the next-hop node for each transmission. Anycasting requires the use of RTS packets to elicit CTS responses from the neighbors before transmission of the packet. We define the Aggregation ID (AID) to associate two packets that can be aggregated. The RTS contains the AID of the transmitting packet and any neighbor that has a packet with the same AID can respond with a CTS. Depending on the application, AID could be any type of data, such as geographic location, instance time, or interested attribute. In this paper we use the measurement timestamp as the AID. Therefore two packets that are generated roughly at the same time can potentially be aggregated. As there could be multiple receivers capable of aggregating the packet, the receivers randomly delay the CTS transmission to avoid CTS collision. Fig. 2 shows the difference between unicasting in 802.11 and randomized CTS response in anycasting. The interference between the neighbors of the sender can prevent multiple CTS transmissions. In DAA, nodes will cancel their CTS transmission if they receive any packet during the random delay. An interference range of more than twice of the transmission range suffices for causing the desired interference.
- **Metric based prioritized CTS:** Using the above approach, packets will converge to a few nodes which we call the aggregation points, and these aggregation points form an independent set. These aggregation points are selected dynamically without explicitly constructing the aggregation tree. However, it is possible that the packets might be forwarded away from the sink. In order to reduce the chance of packets being forwarded away from the sink,



(a) 802.11 based RTS/CTS

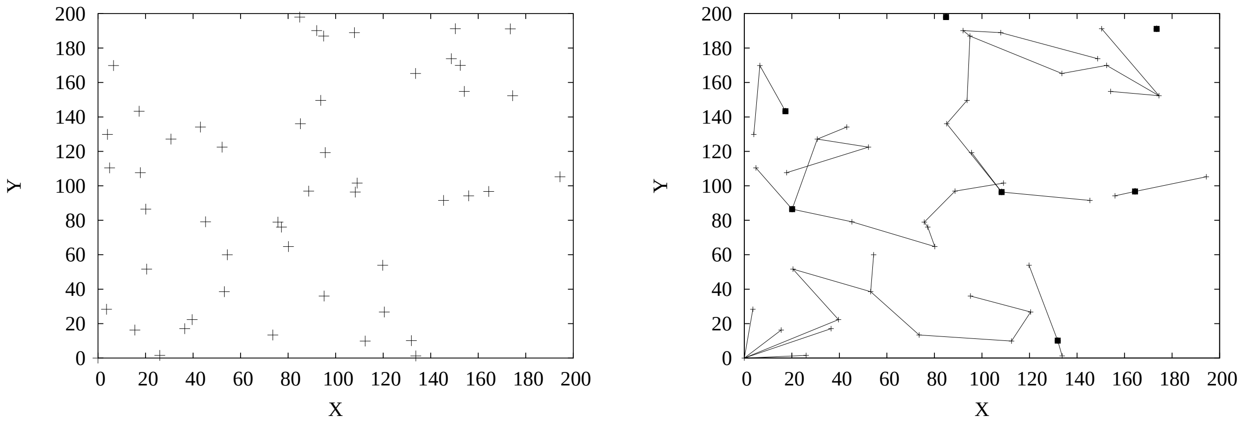
(b) Anycast based RTS/CTS

Fig. 2. **Unicast vs. Anycast:** (a) In 802.11, the receiver sends a CTS immediately after receiving the RTS. (b) In our approach, the receiver sends a CTS with a random delay to avoid collision between nodes sending the CTS. The CTS of receiver 2 has longer delay and hence is canceled after hearing CTS from the receiver 1.

we give higher priority to nodes that are *closer* to the sink than the sender for sending the CTS. Higher priority can be assigned by choosing a shorter random delay before sending the CTS. The metric for proximity could be any metric including geographic distance or number of hops, and is discussed in detail later in this section.

- **DAA on all hops:** To further increase aggregation, we also use the DAA approach rather than unicast while forwarding packets from the aggregation points to the sink. However, in order to forward the packet to the sink using the DAA approach, we enhance the mechanism as follows. Instead of dropping RTS if the nodes do not have the packet for aggregation, they reply with the CTS if they are closer to the sink, but with lower priority than nodes that have packets for aggregation. Therefore, packets are still aggregated when they have the chance to meet, otherwise the packets are forwarded greedily toward the sink.

Fig. 3(a) shows an example network of size $200m \times 200m$ with 50 nodes with the sink at $(0, 0)$. The communication range of each sensor node is $50m$. Fig. 3(b) shows routes taken by packets before they reach the aggregation points (black nodes) where they first fail to get aggregated any further. The result shows that the 50 packets are aggregated to 7 aggregation points (not including the sink), therefore only 7 packets are left in the network after aggregation. Comparing to the 50 packets in the network in the beginning, DAA reduces the packets to only 7 without incurring any overhead for constructing or maintaining a structure.



(a) (b)

Fig. 3. (a) A 200x200 network with 50 nodes. (b) Routes constructed by the spatial convergence anycast protocol during anycast aggregation stage in the 200mx200m with 50 nodes network. The sink is located at (0, 0) and the communication range of each node is 50m.

Theoretically, the average number of aggregation points selected in DAA is $n/(k+1)$ where n is number of nodes generating packets and k is the average degree of nodes. This is intuitive: Consider a node that has k neighbors. It will become an aggregation point only if all its neighbors have sent their packets before itself. The probability that the node sends its packet later than all its neighbors is $1/(k+1)$, therefore the average number of aggregation points is $n/(k+1)$. This means that the number of packets remained in the network is reduced by a factor of $k+1$ automatically, which save a lot of energy if the network is large and many source nodes are far away from the sink.

We now discuss details of the CTS priorities, and the distance metrics.

CTS Priorities: Nodes are assigned different priorities in responding to RTS. The three classes of priorities are as follows:

Class A:The receiver has the packet with the same AID as specified in RTS, and is closer to the sink than the sender.

Class B: The receiver has the packet with the same AID as specified in RTS, but is farther away from the sink than the sender.

Class C: The receiver does not have the packet with the same AID, but is closer to the sink than the sender.

If the receiver does not have the packet with the same AID, and is also farther from the sink than the sender, it does not send a CTS. Corresponding to these three classes of neighbors that can respond to the RTS, three slots are reserved for the CTS packets providing exclusively higher priorities for Class A over Class B, and Class B over Class C (Fig. 4). Nodes in the same class select a mini-slot to send their CTS to avoid collision with other nodes in the same class. In order to further reduce the number of transmissions, we divide Class C into 3 different priorities. Nodes that are on the shortest path to the sink have the highest priority in Class C. Nodes can know this information either by relative physical locations to their neighbors, or by routing protocol indicating they are the next-hop of the sender. Second, nodes that are at least closer to the sink by half of the transmission range than the sender are assigned with priority two, and the remaining nodes in Class C are assigned with priority three. This can reduce the number of transmissions since it takes fewer hops to reach the sink by forwarding packets to farther nodes than closer neighbors when there is no aggregation. Note that the actual transmission time of the CTS could be larger than the mini-slot or slot time. The slots and mini-slots are used to stagger the starting time of CTS transmissions based on the priorities. Based on the assumption of interference between neighbors, we expect only the first CTS transmission to succeed since the others will suppress their transmissions due to the resulting interference.

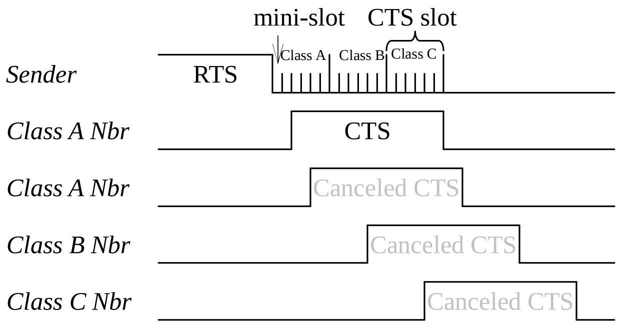


Fig. 4. Nodes with different priorities send CTS in different CTS slots, and nodes with the same priority select different mini-slots. A CTS is canceled if the node overhears another packet (or CTS).

Distance Metrics: In the DAA protocol, nodes need to know whether they are closer to the sink than the sender to set the priority for sending the CTS. This priority is used for selecting the

CTS-slot and also the mini-slot within a CTS-slot. We use *geographic distance* to compare the distance to the sink between two nodes. Nodes have to know their location and also the sink's location. Furthermore, nodes have to know their neighbors' location. Neighbors' location can be either contained in the RTS packet so the receivers can learn the sender's location, or can be exchanged between neighbors during network deployment. Geographic voids and protocols to go around voids have been well studied [31], [32]. The DAA approach can be easily adapted to account for voids. For example the perimeter-mode forwarding approach for dealing with voids [31] can make use of the anycast approach where Class A and Class C can be restricted only to the designated next-hop on the perimeter. Therefore in the perimeter mode the packet can be forwarded to the designated next-hop if it has a packet (Class A), or it can be aggregated at another neighbor (Class B), or it can be forwarded to the designated next-hop without aggregation (Class C). The DAA approach can also be used with other metrics such as the number of hops to the sink. The main difference from the geographic approach is that the number of hops will be used to measure closeness to the sink rather than the geographic distance.

The DAA approach meets the design goals outlined in the beginning of this Section. The DAA approach is used at each hop resulting in aggregation as early as possible on the routes to the sink. As there is no computed structure, event mobility has no impact on the performance of DAA. As transmission links and next-hop nodes are chosen dynamically, DAA is tolerant to interference and node failures, and therefore is very robust even in unreliable networks. However, in the DAA approach packets may not get aggregated if they are spatially separated (more than 1 hop) and if they are forwarded in lock-step by the MAC layer. For such cases, we study the temporal convergence technique for improved performance.

B. Temporal Convergence

The second condition for aggregation requires packets to be present at the same node at the same time. Structure-free aggregation does not guarantee that aggregation will happen even when packets follow the same route. If the order of transmissions do not result in packets meeting temporally at intermediate nodes, the benefit of aggregation may be limited. The order of transmission may be governed by several factors including interference from other flows and interference from the same flow.

Assume that the backoff intervals are much smaller in comparison to the packet transmission time. For such a configuration, packets that are only a few hops apart may get forwarded in lock-step till they reach the sink even though they are on the same route. To illustrate this point, consider a simple topology where all nodes are lined up in a chain as shown in Fig. 5. Suppose the radio signal can interfere with nodes that are two hops away. If node D transmits first, node B and C will remain silent during the transmission. Therefore no nodes will contend for the channel with A. Although C will send CTS packet and the channel will not be idle for node A, node A will only backoff for a short period, which is shorter than a packet transmission time, and will sense the channel as idle after that. Since there is no contention, node A will send its packet, and its packet will not be aggregated with other packets from upstream nodes. Note that when packets are more than one hop apart or when packets follow the same route, the DAA approach is ineffective in improving aggregation.

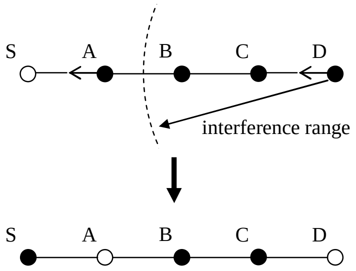


Fig. 5. Packet from D and A can hardly be aggregated if nodes forward packet in lock-step.

Deterministically assigning the waiting time to nodes such that nodes closer to the sink wait longer before transmitting can avoid this problem. However nodes have no knowledge of the event size and location, and don't know their relative position compare to other nodes sensing the event. The only information that a node knows is their distance to the sink. Therefore they can only set the delay inversely proportional to their distance to the sink. This results in a fixed delay for all packets wherever the event is, and the delay will be proportional to the size of the network, which would be intolerable in large network deployment.

Therefore we propose Randomized Waiting (RW) at the source for each packet to introduce artificial delays and increase temporal convergence. Each node generating a new packet to transmit, delays it by an interval chosen from 0 to τ, where τ is the maximum delay. In Fig. 5, if node A chooses a higher delay than node D and nodes B and C have lower delays than A or D, node D's packet may be aggregated at node A if the difference between the delays of A

and D is greater than the transmission time from node D to A. Notice that with Randomized Waiting, it is possible that the packets may be transmitted out of order if the data sampling time is smaller than τ.

The optimum value of τ depends on the event size, i.e. the maximum hop distance between nodes that are generating packets. If the maximum number of hops increases, the maximum delay should increase such that the difference between the delay chosen by two nodes increases. If the difference between two delays are too close, packets will not be aggregated even if the downstream nodes have higher delay because the transmission time will be greater than the delay difference. However if the maximum delay is too large, the end to end packet transmission latency will be too high. If the application is not delay tolerant, a low value of τ will need to be used which can not reap the benefits of this approach. We explore this trade-off in Section V.

IV. Performance Analysis

In this section we model the performance in terms of the total number of packets transmitted in the network when both Data-Aware Aggregation and Randomized Waiting are used. An alternate combinatorial analysis technique is shown for a special case that also validates the analysis. Results from simulations match the analysis results closely.

A. *Expected Number of Transmissions*

In this section we compute the expected number of transmissions when both spatial and temporal convergence techniques are used together. In a network where all nodes have data to send, if nodes can cooperatively construct an aggregation tree and transmit packets starting from leaves to the root, there are $n-1$ transmissions where n is the number of nodes in the network since there are $n-1$ edges in the constructed tree.

To analyze the expected number of transmissions in a structure-free network, first we compute the probability that a packet will be aggregated. We assume that each node in the network has a packet to transmit. Each node picks a random delay for every packet that it originates. If downstream nodes have higher delay than the upstream nodes, packets can be aggregated at the downstream nodes. To simplify the analysis, nodes only send packets to nodes that are closer to the sink, and we do not consider the transmission delay that may result in fewer aggregations.

Let Y be the random variable representing the number of hops a packet has been forwarded when it is aggregated. Let $d_{v_h} = x$ be the normalized random delay between 0 and 1 chosen by v_h where v_h is a node that is h-hops away from the sink. Consider a network where each node has an average of k choices for downstream nodes. A packet can be forwarded i hops and be aggregated only if a) the packet is forwarded through $i-1$ nodes and all these nodes have lower delay than the sender, and b) the i-th node has higher delay than the sender. Therefore, for a node that is h hops away from the sink,

$$P(Y=i) = \begin{cases} x^{(i-1)k} \times (1-x^k), & \text{if } 0 < i < h \\ x^{(i-1)k}, & \text{if } i = h \end{cases} \tag{1}$$

The expected value of Y when the delay is x for node v_h is:

$$\begin{aligned} E[Y|d_{v_h} = x] &= \sum_{i=1}^{h} i \times P(Y=i) \\ &= \left(\sum_{i=1}^{h-1} i \times x^{(i-1)k}(1-x^k)\right) + h \times x^{(h-1)k} \\ &= \left(\sum_{i=1}^{h-1} i \times x^{(i-1)k}\right) \\ &\quad - \left(\sum_{i=1}^{h-1} i \times x^{ik}\right) + h \times x^{(h-1)k} \\ &= \sum_{i=0}^{h-1} x^{ik} \end{aligned} \tag{2}$$

Because x is uniformly chosen between 0 to 1, the expected value $E[Y]$ is therefore

$$\begin{aligned} E[Y] &= \int_0^1 E[Y|d_{v_h} = x]\, dx \\ &= \int_0^1 \left(\sum_{i=0}^{h-1} x^{ik}\right) dx \\ &= \left[\left(\sum_{i=0}^{h-1} \frac{x^{ik+1}}{ik+1}\right)\right]_0^1 \\ &= \sum_{i=0}^{h-1} \frac{1}{ik+1} \end{aligned} \tag{3}$$

Using this expected value, we can calculate the expected number of transmissions in the network as the summation of all expected number of transmissions of nodes from hop 1 to n/k (assume that the n nodes are uniformly distributed and all nodes at the same hop distance have the same k downstream nodes, i.e. each level has k nodes, therefore the maximum hop number is n/k in average), and is:

$$\sum_{h=1}^{n/k} k \sum_{i=0}^{h-1} \frac{1}{ik+1} \quad = (n+1)H_k(\tfrac{n}{k}) - \tfrac{n}{k} \tag{4}$$

where $H_k(n) = \sum_{i=1}^{n} \frac{1}{(i-1)k+1}$ is the summation of a harmonic sequence.

B. Alternate Analysis

The above analysis assumes an uniform distribution for x in [0,1]. However the result is independent of the distribution. Using an alternate combinatorial technique for the special case of $k = 1$, we obtain the same result. We present the alternate technique (only for the special case) for validating our analysis and for the sake of completeness.

Consider a chain topology of nodes v_0 to v_n when node v_0 is the sink and all nodes have data to report to the sink. Picking a number from 0 to 1 for each node is equivalent to choosing a random permutation corresponding to the order of transmissions. As shown in Fig. 6, for a packet generated at v_n to be forwarded h hops, node v_n must transmit later than all nodes within $h-1$ hops, and earlier than the node at h hops away. It is equivalent to randomly assigning n distinct numbers, s_{v_1} to s_{v_n}, to these n nodes as their orders of transmissions such that among nodes v_{n-h} through v_n, $s_{v_{n-h}}$ is the largest number and s_{v_n} is the second largest number. There are $\left(\begin{array}{c} n \\ h+1 \end{array}\right)$ possible combinations of selecting $h+1$ numbers out of n numbers. Among these selected $h+1$ numbers, there are $(h-1)!$ possible orderings such that $s_{v_{n-h}}$ is the largest number and s_{v_n} is the second largest number. The number of possible orderings of the rest of $n-(h+1)$ nodes is $(n-(h+1))!$ Therefore the probability for a packet to be forwarded h hops is

$$\frac{\left(\begin{array}{c} n \\ h+1 \end{array}\right) \times (h-1)! \times (n-(h+1))!}{n!} = \frac{1}{h(h+1)}$$

If the packet travels to the sink without any aggregation, then all nodes on the route must have lower delay than the source. For n nodes the fraction of permutations where a given node

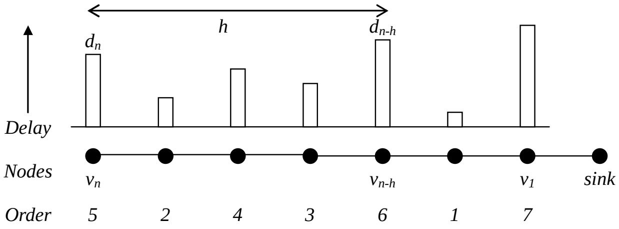


Fig. 6. The order of transmissions when nodes choose a randomly delay.

has the highest delay are $\frac{1}{n}$. Therefore,

$$P(Y = i) = \begin{cases} \frac{1}{i(i+1)} & \text{if } 0 < i < n \\ \frac{1}{n} & \text{if } i = n \end{cases} \tag{5}$$

and the expected value $E[Y]$ for node v_n is:

$$\begin{aligned} E[Y] &= \sum_{i=1}^{n} i \times P(Y = i) \\ &= \sum_{i=1}^{n-1} i \times \frac{1}{i(i+1)} + n \times \frac{1}{n} \\ &= \sum_{i=1}^{n} \frac{1}{i} \qquad (6) \end{aligned}$$

and the expected number of transmissions in the chain topology network is

$$\begin{aligned} \sum_{h=1}^{n} \sum_{i=1}^{h} \frac{1}{i} &= (n+1)H_1(n) - n \qquad (7) \\ &= (n+1)(\ln n + O(1)) - n \qquad (8) \\ &\approx n \ln n \qquad \text{if } n \to \infty \qquad (9) \end{aligned}$$

We observe that this expression matches Equation 4 for $k = 1$, validating the previous analysis. This alternate analysis also confirms that the expected number of packets is independent of the distribution used to choose the random number.

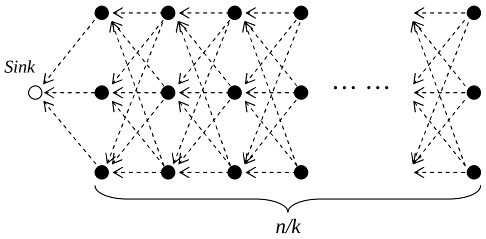


Fig. 7. A network topology where each node has k=3 choices for downstream nodes.

C. Comparison with Simulation Results

To compare the analytical results with simulations (using $ns2$) we use the network topology shown in Fig. 7. Each node has three downstream nodes within its transmission range. In this simulation we only allow nodes to send packets to one of the 3 downstream nodes in the next column, and do not send packets to the nodes in the same column or nodes away from the sink. This corresponds to $k = 3$ in our analysis. We use $\tau = 2$ seconds for the maximum random delay for delaying packet transmission. Including the overhead of anycast for 50 byte packets this is approximately 50 packets transmission time.

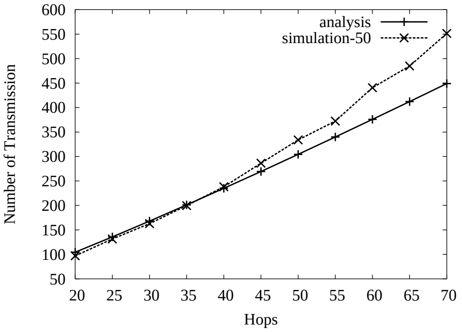


Fig. 8. The number of transmissions vs. the network size for the analysis and simulation results.

Fig. 8 shows the results of simulations and analysis. It shows that the analysis results match the simulation results when the network size is less than 40 hops. As the hop count increases, the number of transmissions measured from the simulations increases faster than the predictions of the analysis. This difference is due to the absence of a model for transmission delay in our analysis. As the number of hops increases the transmission delay also increases. Therefore, a packet may not get aggregated at the downstream node which has higher delay due to nonnegligible transmission delay. As this effect increases with increasing number of hops, the discrepancy accordingly increases. Note that with a larger value of τ the simulation results can be brought closer to the analysis for a wider range of number of hops. Although the delay is introduced only at the source, the resulting end-to-end delay may not be acceptable to the application if τ is very high.

V. Performance Evaluation

In this section we evaluate the performance of our protocol. In order to justify the design of the structure-free approach, we compare our protocol with a structure approach. We construct an aggregation tree for each instance of measurement in advance, and the root of these trees are located at the center of the event. Nodes delay their transmission according to their height in the tree, therefore packets are transmitted starting from the leaves to the root to achieve highest number of aggregation. Parent nodes will transmit their packets once they receive packets from all their children, or when the delay timer expires. This approach should have best performance since nodes know how long to wait for their children and to where they should send their packets to achieve maximum aggregation. Notice that these trees are constructed implicitly assuming the network topology and the movement of the events are known in advance. It does not consider the cost of constructing and maintaining the tree. We also include the opportunistic aggregation in the comparison.

The protocols that evaluated in this section are listed below:

1) **Opportunistic Aggregation (OP)**. Nodes send their packets along the shortest path to the sink immediately when they get the measurements. Packets are aggregated opportunistically.
2) **Randomized Waiting (RW)**. Nodes send their packets along the shortest path to the sink. Packets are delayed at the source and are aggregated opportunistically.
3) **Data Aware Anycast (DAA)**. Nodes use spatial convergent anycast to aggregate packets without delaying at the source as described in section III-A.
4) **Data Away Anycast with Randomized Waiting (DAA+RW)**. Both DAA and RW approaches are used.
5) **Aggregation Tree (AT)**. The structured approach described in the beginning of this section. According to the simulation, we set the delay timer of a node to be 0.64 seconds, which

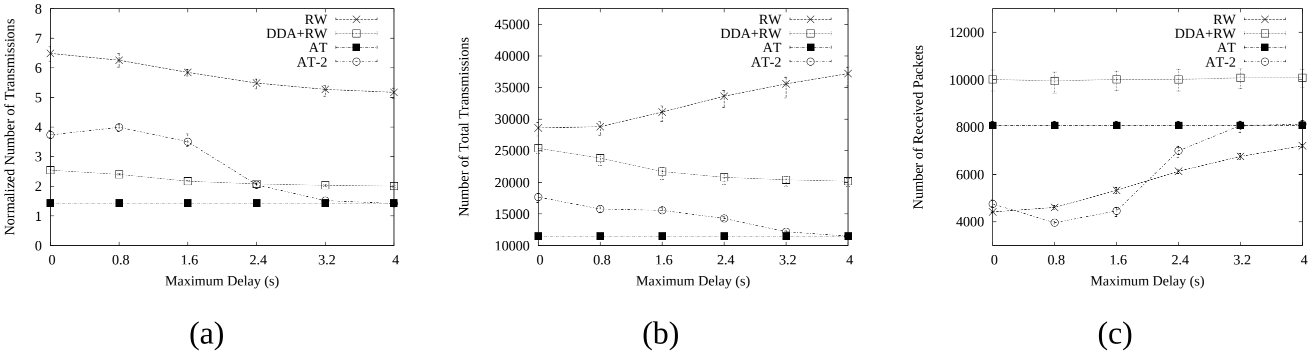
is about 16 packets transmission time, longer than its children, and the maximum delay do not exceed 8 seconds. Therefore nodes with hight 1 (leaves) do not delay, nodes with height 2 delay 0.64 seconds, nodes with height 3 delay 1.28 seconds...etc, to get the best performance. However, nodes will send their packets earlier if they get packets from all their children before the delay timer expires.

We use the *ns2* network simulator to evaluate the performance of these protocols. The RTS/CTS packet formats of 802.11 MAC are modified to incorporate the anycasting capability. The RTS contains an extra field of Aggregation ID and CTS packet contains an extra field of the address of the CTS sender. In all scenarios there is only one sink in the network and geographic location is used to compare the distance from nodes to the sink. We assume that nodes know their neighbors' location.

The network is a $1000m \times 1000m$ square region with grid topology. The sink is located at one corner of the network. The data rate of the radio is $38.4Kbps$ and the communication range is slightly longer than $50m$. An event moves in the network using the random way-point mobility model for 400 seconds. Nodes generate packets with 50 bytes payload, and send packets to the sink every 5 seconds. For an event size of radius of $200m$ with $25m$ as the distance between two nodes, there are about 200 nodes generating packets at the same time (or 50 nodes generating packets if the distance between two nodes is $50m$). Unless otherwise mentioned, the inter-node separation is $30m$, the event moving speed is $10m/s$ with a pause time of 0 second, the radius of the event is $200m$, and the maximum delay (τ) for RW and DAA+RW approaches is 3.2 seconds. All simulation results are based on 5 different mobility scenarios (each for 400 seconds). The minimum and maximum value obtained are also drawn in all graphs.

The normalized number of transmissions is used as the metric to compare different protocols. Normalized number of transmissions represents how effective a protocol is in aggregating packets and is *(Number of transmissions in the network)/(Number of Contributing Sources)*. The *Number of Contributing Sources* is the effective information that are generated by all sources in the network and are aggregated at the sink. The number of aggregation can not tell if a protocol performs well since packets might be forwarded many hops before being aggregated, and reducing the number of transmissions is a key design goal. The number of transmissions will be lower if more packets are aggregated early on their routes to the sink. However the number of transmissions could be low if a lot of packets are dropped. Therefore we use the normalized number of transmissions as the metric to compare different protocols.

A. Maximum Delay



(a) (b) (c)

Fig. 9. The simulation results for different maximum randomized waiting time. (a) Normalized number of transmissions. (b) Total number of transmissions in the network. (c) Number of received packets.

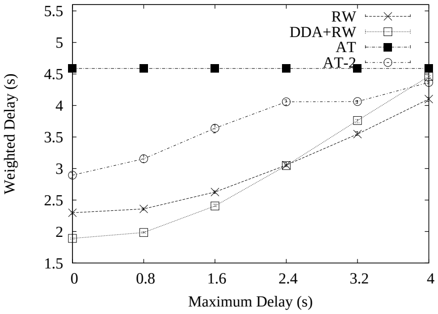


Fig. 10. The end-to-end transmission delay v.s. maximum delay.

First we evaluate the performance of the RW approach for achieving temporal convergence. Using the default scenario ($30m$ inter-node separation, $200m$ event radius and $10m/s$ event speed), there are around 140 nodes generating packets at the same time and each node has about 8 neighbors in its communication range. We vary the maximum delay τ from 0 (no delay) to 4 seconds. Fig. 9 shows the results. In Fig. 9, AT uses 8 seconds as the maximum delay, i.e. if the height of the tree is higher than 13, than the node with height larger than 13 will only delay 8 seconds (the delay of each node is $0.64 * (height - 1)$). AT-2 is AT approach with the maximum delay specified as the X-axis. OP and DAA are not shown in these graph, but they are just RW and DAA+RW approaches with $\tau = 0$.

As we can see, the normalized number of transmissions decreases as the maximum delay increases. At $\tau = 100$, the DAA+RW is 21% lower than DAA (DAA+RW with delay 0), and RW is 20% lower than OP. The structure approach AT, as predicted, has the lowest normalized number of transmission. AT is 29% lower than DAA+RW in terms of normalized number of transmission. However AT requires the time and overhead of constructing the tree, which may degraded its performance further, and is not shown in the graph. AT-2, the structure approach with different maximum delay, does not perform better than DAA+RW approach when the maximum delay is less than 2.4 seconds. This shows that structure approach are very sensitive to the waiting time as we described in Section I. At worst case, AT-2 has 300% more normalized number of transmission than AT. However with higher delay, the end-to-end delay is also higher for AT. (Fig. 10). For DAA+RW approach, although with higher values of τ, the normalized load can be further reduced, beyond $\tau = 1.6$ the reduction is marginal.

Fig. 9(b) and 9(c) show the number of transmissions and receptions of packets for different τ. We can see that the AT has the lower number of transmission, which is only 57% of DAA+RW. However, DAA+RW has higher number of received packets than AT. This is counter-intuitive since AT delays packets transmission nicely, it should have less packet dropping rate than DAA+RW. However, tracing into the simulation logs we find that the packet dropping rate is very high in AT. DAA+RW has about 7 packets loss which counts for around 150 effective information, while AT has about 200 packets loss which counts for about 2100 effective information. We believe that this is because in the AT approach, nodes have only one choice, their parent, to send their packets to. It is a converge cast which may cause a lot of contention. However in DAA and DAA+RW, nodes have multiple choices as next-hop, and packets tend to be forwarded away from each other (packets that are close will be aggregated), which reduce the contention. Therefore DAA and DAA+RW have less packet loss and achieve higher number of received packets.

The other interesting phenomenon is, as DAA+RW achieves lower number of transmission than DAA and it decreases as τ increases, RW has opposite behavior as DAA+RW does. RW has higher number of transmissions than OP, and it increases as τ increases. The increase of number of transmissions in RW is because the decrease of the packet dropping rate. We can see that the number of received packets increases in RW as τ increases in Fig. 9(c). As fewer packets are dropped, more packets are transmitted in the network, and they contribute to the increase of number of transmission. The reason that packet dropping rate decreases as τ increases is, as τ increases, packets are more likely to be transmitted at different time, and the collision is reduced. The higher packet dropping rate does not happen in DAA shows that DAA does achieve packet aggregation in the beginning, therefore it reduces the number of packets remained in the network and lower the probability of collision.

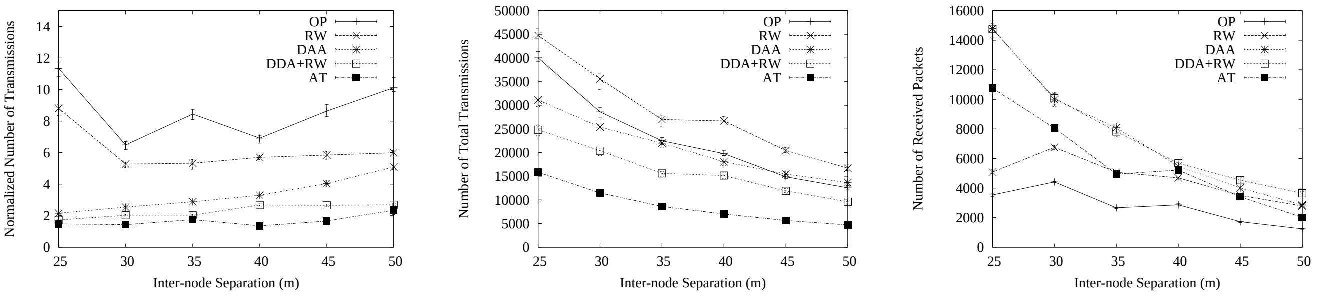
Fig. 10 shows the weighted delay for different τ. The *weighted delay* is the average delay experienced by a packet reaching the sink weighted by the number of contributing sources for that packet. For example, suppose at time 2 the sink receives a packet generated at time 0 and this packet contains 3 aggregated packets. At time 8, the sink receives another packet containing 4 aggregated packets which are generated at time 5, the weighted delay is $((2-0)*3+(8-5)*4)/(3+4) = 18/7$. Therefore, as τ increases, the average weighted delay also increases.

We can see that the structure approaches have higher delay than the structure-free approaches. The main reason is that nodes have to wait for all their children before the delay timer expires, and when the aggregated packet reaches the sink, the delay is longer than average delay in structure-free approaches. In some applications, such as intrusion detection, the sink might want to get coarse data very soon to reduce the detection latency, and later get more concrete information about the event. AT can not achieve this since it is supposed to wait packets from all their children before transmitting. For these applications, DAA+RW provides a better trade-off between energy and delay. All protocols reduces the normalized number of transmissions by introducing the waiting delay at the expense of increasing the end-to-end delay. For applications that are not tolerant to delay, DAA still provides good performance with lowest delay.

B. Node Density

In this section we evaluate the performance of our protocols in networks with varying node density. With higher node density, packets are more likely to meet and get aggregated in DAA. Fig. 11 shows the results for different protocols for different node densities.

In general, the normalized number of transmissions decreases as the node density increases as predicted. DAA and DAA+RW perform much better than OP and RW approaches. At the highest node density, DAA+RW improves the normalized number of transmission by 73% compare to the OP approach. AT still has the lowest normalized number of transmissions among all protocols, but is very close to DAA+RW. AT is only 14% lower than DAA+RW at the highest node density



(a) (b) (c)

Fig. 11. The simulation results for different node densities. (a) Normalized number of transmissions. (b) Total number of transmissions in the network. (c) Number of received packets. The X-axis is the distance separating two adjacent nodes.

simulation ($25m$ inter-node separation).

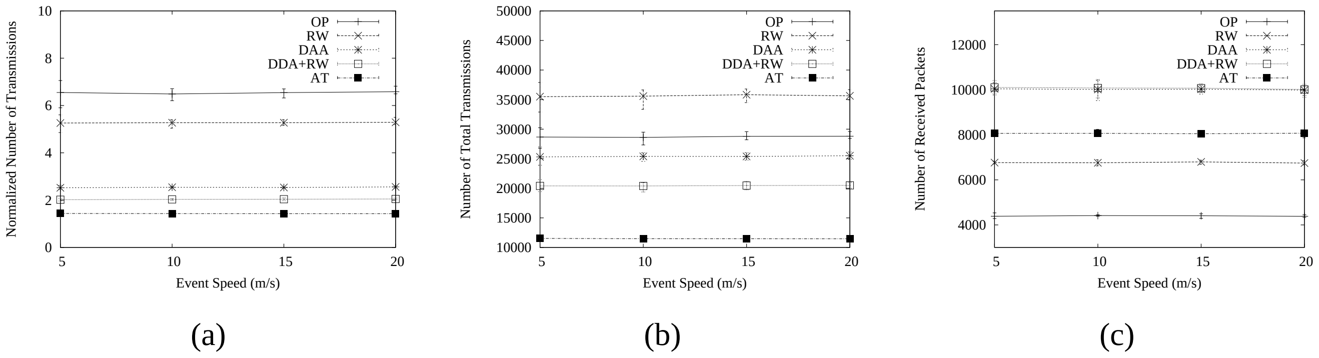
DAA and DAA+RW still have the highest number of received packets due to their ability to aggregate packets at very beginning, and scatter packets away from each other to reduce the contention and decrease the dropping rate. The reason that the results of OP and RW are not smooth is that packet dropping rate is very high in these two protocols. About 67% to 76% packets are dropped during transmissions, compared with less than 4% to 5% dropping rate for DAA and about 26% for AT. Because the total number of transmissions includes packets that did not reach the sink, and the normalized number of transmissions is averaged over the number of received packets, the results are very sensitive to the number of dropped packets.

C. Event Speed

Fig. 12 shows the results for different event moving speeds. We vary the event speed from $5m/s$ to $20m/s$. From the figure we can see that the results remain quite steady at different speeds. As they do not create any structure for aggregation, mobility has little influence on the performance. AT maintains the same performance across all speeds because we do not take the overhead of constructing the structure into consideration. We believe that when considering all the tree construction head, the performance of AT will degrade as the speed increases.

D. Event Size

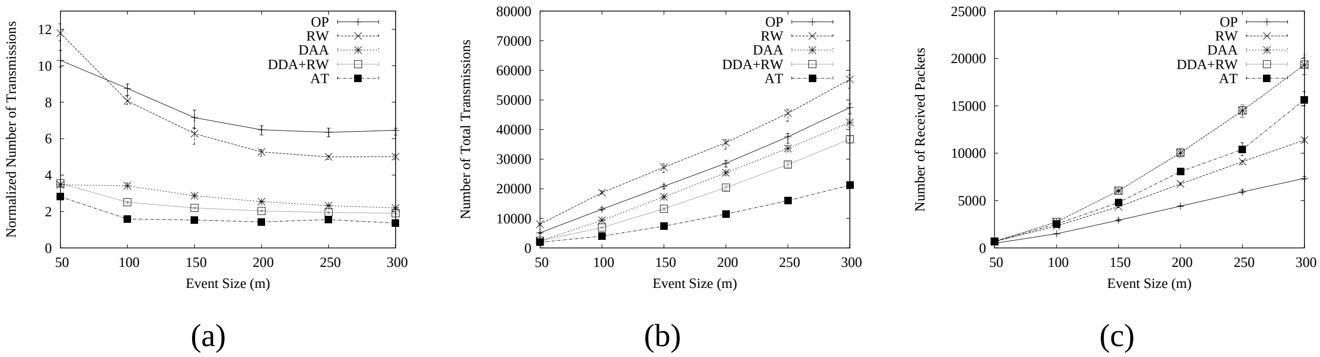
Fig. 13 shows the results for different event size. We vary the radius of the event size from $50m$ to $300m$. The number of transmitted and received packets increases as the event size



(a) (b) (c)

Fig. 12. The simulation results for different event moving speeds. All protocols' performance are steady since we construct the tree in advance for AT, and others are structure-free approaches which does not be affected as the speed changes.

increases because more nodes are sensing the event. All protocols improve the performance as the event size increases, because when the event size increases, more nodes are sending the packets, and packets have more chances to be aggregated, therefore reduce the normalized number of transmission. With smaller event size, the performance of OP and RW approaches degrades quickly, but the performance of DAA and DAA+RW approaches only decrease slightly. This shows that DAA does get benefit from aggregating packets generated from nodes closed to each other, therefore even though it is structure-free approach, it still performs better than opportunistic aggregation.



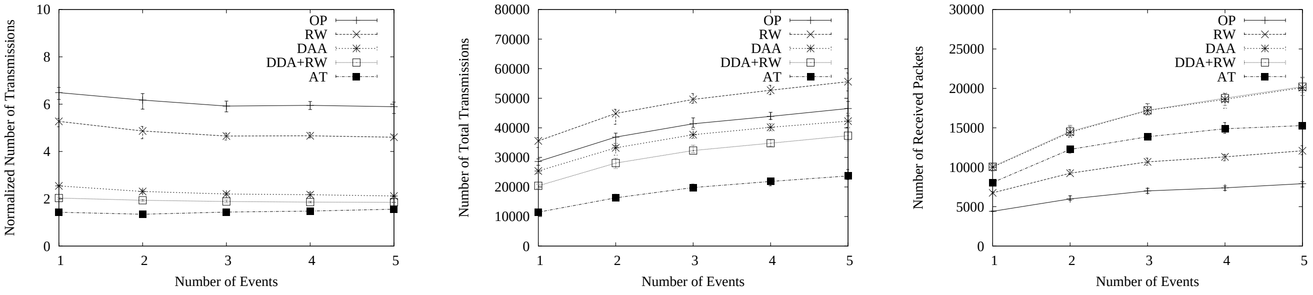
(a) (b) (c)

Fig. 13. The simulation results for different event size. As event size increases all protocols perform better because more packets are aggregated.

E. Number of Events

Fig. 14 shows the results for different number of events in the network. The results are very similar to the evaluation for event size. With more number of events, more packets are generated, and they are more likely to be aggregated.

Notice that the number of generated packets does not increase proportionally compare to the increase of the number of events. This is because within a $1000m \times 1000m$ networks with the $200m$ radius of a event, most of the events are overlapped and the nodes only generate 1 packets no matter how many events are within its sensing range.



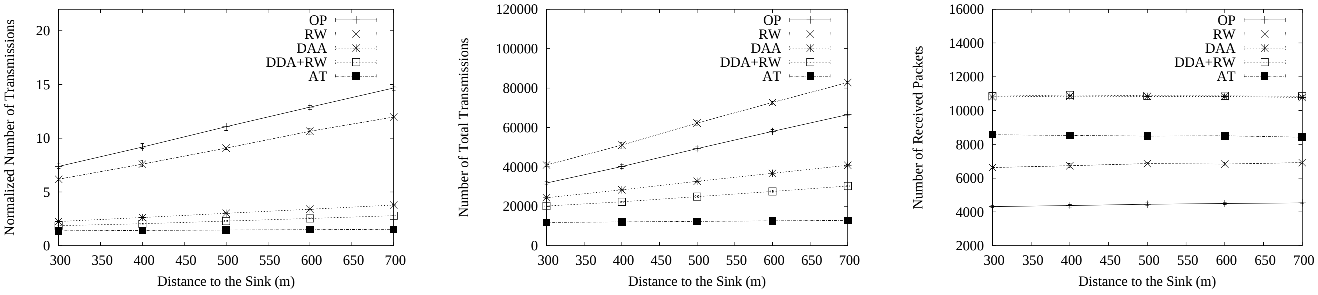
(a) (b) (c)

Fig. 14. The simulation results for different number of events.

F. Distance to the Sink

Unlike AT that can aggregate all packets into one node (optimally) because it is a structure approach, DAA and DAA+RW are unlikely to aggregate all packets into only one node. As we described before, there will be $\frac{n}{k+1}$ packets left in the network in average after DAA approach, where n is the number of nodes generating packets and k is the average number of neighbors a node has. As more packets are left in the network, the cost of forwarding these packets are higher, and the increase is more significant as the distance of the event to the sink is longer.

In this section we evaluate the impact of the event distance to the sink on our protocols. We restrict an event to move within a $200m \times 200m$ square centered at $(300, 300)$ to $(700, 700)$, and observe the performance of the protocols at different distances. Fig. 15 shows the results of these simulations. As we predicted, when the distance to the sink increases, all structure-free protocols have higher number of transmissions. From the closed distance to the farthest distance, OP approach almost doubles its normalized number of transmission, and DAA+RW approach is 1.6 times higher. Only AT remains the same across all scenarios. This suggests us that some semi-structure approaches might be helpful to reduce the number of packets left in the network and improve the performance over structure-free approaches. This is considered as future work and is not in the scope of this paper.



(a) (b) (c)

Fig. 15. The simulation results for different distance between the event and the sink. When the event is farther, the performance drops for structure-free approaches.

G. Aggregation Ratio

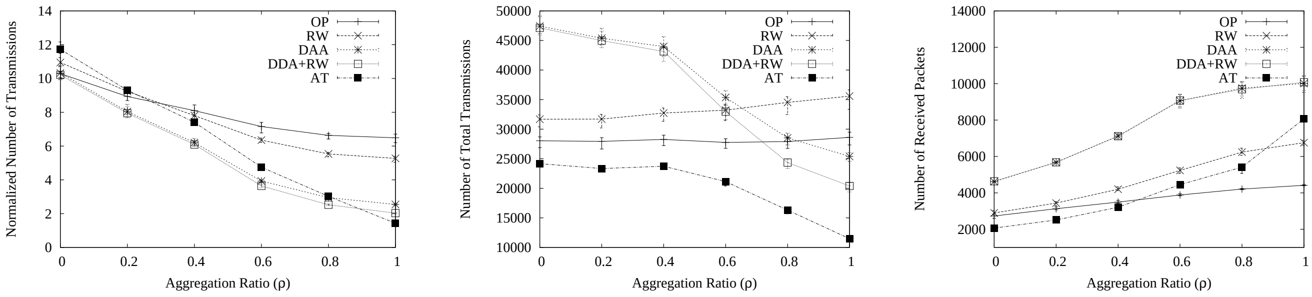
All simulations in previous sections focus on perfect aggregation, that is, no matter how many packets are aggregated, they can be aggregated to one packet. In this section, we study the impact of aggregation ratio for different protocols. When a node senses an event, it will generate a packet with 50 bytes payload. We use a simple aggregation function to aggregate packet: The size of the packet after aggregation is $max\{50, n \times (1-\rho)\}$ where n is the number of effective information and ρ is the aggregation ratio. $\rho = 1$ stands for perfect aggregation. The maximum payload of a packet is set to 200 bytes. Therefore two packets may not be aggregated even if they meet at the same node at the same time if the aggregated size is greater than 200 bytes. In DAA and DAA+RW, nodes should be able to distinguish if they can achieve aggregation when the receive a RTS from its neighbors. Therefore we add a field to RTS to specify how many effective information are contained in the data packet to let node know their priority of replying a CTS.

Fig. 16 shows the results of different aggregation ratio for different protocols. The first thing we notice is AT does not perform well in scenarios with aggregation ratio other than 1. The number of packets received in AT drops very quickly as the aggregation ratio decreases. It becomes the lowest when the aggregation ratio is less than 0.4. The simulation logs show that the packet dropping rate is extremely high in AT when the aggregation ratio is not 1, and it increases as the aggregation ratio decreases. This is due to the following reason. As the aggregation ratio decreases, more packets will remain in the network because they may reach the maximum payload and can not be aggregated anymore. In AT, when packets converge to the aggregation root, they become larger because they aggregate more packets. We observe that when packet reaches nodes with height 4, they may not aggregate packets anymore, therefore more packets will be forwarded to the root. More packets with larger size make the contention even worse. Furthermore, these packets contains more effective information. When they are dropped, more effective information are dropped. This phenomenon does not happened in other protocols because they are not converge-cast to a node. Although OP and RW approaches are converge-cast to the sink, the effect is not so severe as AT because the converge-cast happens in entire network, not within a few hops as in AT.

In this simulation we observe that, structure approaches may not perform well if the aggregation is not perfect. Here we do not consider the cost of constructing and maintaining the structure yet. By incurring the overhead of the structure approaches, the performance would be even worse. On the other hand, as the aggregation ratio decreases, AT can not aggregate all packet into one packet anymore. There will be more packets left in the network. Therefore we conclude that it is not necessary to find a structure to let packets be aggregated to one node. DAA and DAA+RW does not incur the overhead of constructing the structure, and naturally lower the impact of more packets left in the network as the aggregation ratio increases because they only aggregate packets that are close to each other. We believe that the structure-free approach can perform better than structure approach in these scenarios.

VI. Experiment Evaluation

We have implemented the Randomized Waiting (RW) approach over unicast on the Kansei test-bed [33]. The Kansei test bed is composed of 195 stargates [34] arranged in a 15×13 grid. Each stargate is attached with a XSM/Mica2 mote [35] through the serial port. The stargates are used to create a wired network facilitating the reprogramming of XSMs and job dispatch. Each attached XSM runs TinyOS [36]. It is equipped with multiple sensors and one CC1000 radio



(a) (b) (c)

Fig. 16. The simulation results for different aggregation ratio. DAA and DAA+RW perform better than AT when aggregation ratio less than 0.8.

module [37]. This section introduces our experimental methodology and exhibits the advantage of using the RW approach for improving the performance of data aggregation.

A. Experiment Design

The experimental topology is a 5-hop network with 6 XSM placed in a row (Fig. 17). One XSM acts as the sink and the other 5 XSMs generate packets when they detect an "event". We assume that all sensor nodes detect each event almost at the same time, and use perfect aggregation to combine any number of packets with the same AID (time-stamp in real deployed application) into a packet.

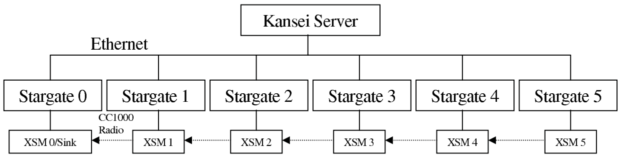


Fig. 17. Experiment environment structure

To mimic the event to trigger packet generation from XSM's at almost the same time, an event broadcast program runs on the Kansei server that connects directly to all the stargates through Ethernet. This program is also in charge of broadcasting statistics query messages to all stargates. There is another program running on these stargates to forward the event message and statistics query message to XSM attached to them, and to receive the statistics report message from the XSM and store it in the disk.

Each XSM waits for an event message, at which time it generates a 36-byte Active Message, inserts the current AID and its node ID into the data field, and puts it into the sending buffer. The XSM delays sending it by starting a timer for a random period of time, and sends the packet to the MAC layer only when the timer expires. If an XSM receives a packet with the same AID as the delayed packet before the timer expires, it aggregates the received packet into the delayed packet. Otherwise it just puts the received packet in the sending buffer. When the timer fires, the application calls the *SendMsg()* method of *GenericComm* to send the packet to the MAC layer, and never tries to retrieve the packet even if a new incoming packet has the same AID as that packet.

The RW approach is compared to opportunistic aggregation (OP) in unicast, which is also implemented on TinyOS. When a node running OP receives an event message from the attached stargate, it generates a packet and sends it to the *GenericComm* component without any delay. When it receives a data packet, it does not try to fetch the packet from the MAC layer for aggregation. Thus the only chance for opportunistic aggregation to happen is when a node has packet pending at the MAC layer and two or more packets with the same AID are received.

B. Experiment Results

To compare the RW approach with opportunistic aggregation in unicast (denoted by maximum delay of 0 in graphs), the experiments are conducted for different value of maximum delay. Since the link quality is not perfect and packets might be dropped, we use two normalized metrics to compare RW approach with OP approach, *ratio of total transmissions vs. aggregations* and *ratio of total receptions vs. aggregations*. Here the aggregation means the the number of different sources contributing to the information of this packet. If the sink receives 10 packets and each of which contains the information from 5 source nodes, then the amount of information is counted as $5 \times 10 = 50$. If a packet contains messages from more sources than another one, it has more information.

The results are shown in Fig. 18 and Fig. 19. Each point in the graph is obtained by generating 500 events. Comparing the opportunistic aggregation with RW, we can see the former has more packet transmissions but the information received by the sink is less ((Fig. 19). The reason is that in opportunistic aggregation, few packets get aggregated, thus more packets are injected into the network, resulting in more contention losses. Note that the maximum delay can be as low as $0.2s$ to achieve comparable performance as large delays ($> 1s$), which means that we do not necessarily incur high end-to-end delay by using the RW approach. Further study of the optimal delay time will be part of our future work.

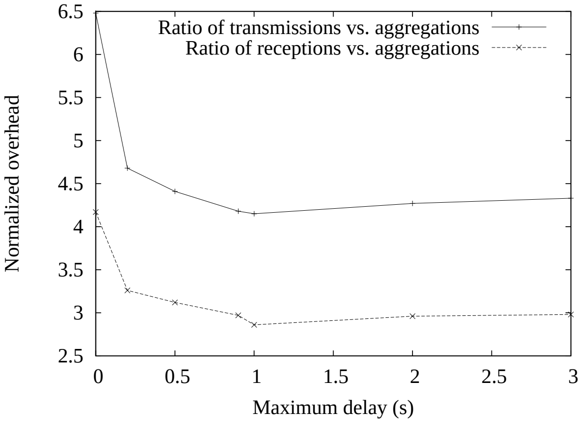


Fig. 18. Normalized overhead for RW approach on 6-node chain topology for OP (0 delay) and RW approaches.

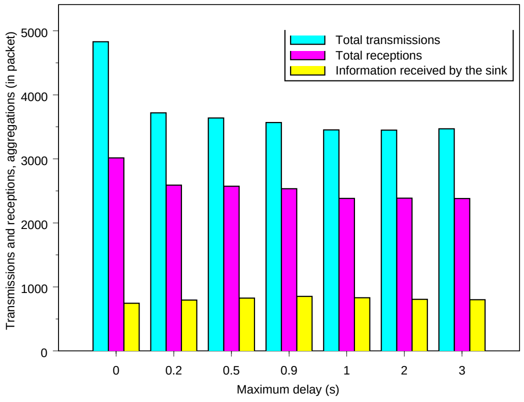


Fig. 19. Statistics of total transmissions, receptions and aggregations. 0 delay is the OP approach while other delays are RW approaches with different delays

We have also gathered the information aggregated at each intermediate node. When a packets arrives at an intermediate node, it may be aggregated to another packet or forwarded directly. Obviously higher aggregation chance is desired. Fig. 20 shows the information aggregated at each intermediate node (node ID from 1 to 4, node 0 is the sink, and node 5 is the most distant source). It is clear from the graph that the aggregation rarely happens in opportunistic aggregation, while the RW approach achieves much more aggregation at intermediate nodes.

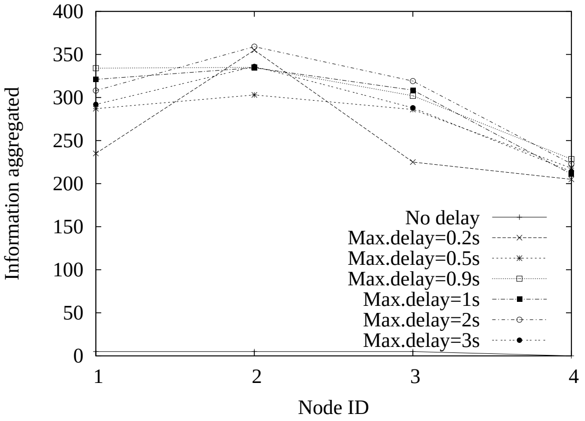


Fig. 20. Information aggregated at intermediate nodes

VII. Conclusion

Research on data-aggregation protocols has focused on tree-based or cluster-based structured approaches. In this paper we proposed techniques for data aggregation that do not use any explicit structures. Efficient aggregation requires packets to meet at the same node (spatial convergence) at the same time (temporal convergence). For spatial convergence we proposed a MAC layer anycast based approach called Data-Aware Anycast (DAA). For temporal convergence we proposed Randomized Waiting (RW) at the application layer at the source of the packet. We model the network load generated by the combined DAA with RW approach and show that the predictions of the analysis match closely with the simulation results. We define the normalized network load as the number of packets transmitted in the network normalized by the total amount of information (number of nodes whose packets reached the sink with or without aggregation) delivered to the sink. Using extensive simulations we show that the combined DAA with RW approach can improve the normalized load by as much as 73% compare to opportunistic aggregation, and it performs better than structure approach when the aggregation function is not perfect. Based on the experimental study with 6 XSM sensor nodes we observe that RW can significantly reduce the normalized overhead in terms of number of transmissions. This shows that structure-free data aggregation techniques have great potential for event-based applications.

REFERENCES

[1] W. 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*, vol. 2, January 2000.

[2] ——, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” in *IEEE Transactions on Wireless Communications*, vol. 1, October 2002, pp. 660–670.

[3] S. Lindsey and C. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems,” in *Proceedings of IEEE Aerospace Conference*, vol. 3, March 2002, pp. 1125–1130.

[4] J. Wong, R. Jafari, and M. Potkonjak, “Gateway placement for latency and energy efficient data aggregation,” in *29th Annual IEEE International Conference on Local Computer Networks*, November 2004, pp. 490–497.

[5] W. Zhang and G. Cao, “DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks,” in *IEEE Transactions on Wireless Communications*, vol. 3, September 2004, pp. 1689–1701.

[6] ——, “Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks,” in *Proceedings of INFOCOM 2004*, vol. 4, March 2004, pp. 2434–2445.

[7] C. Intanagonwiwat, D. Estrin, and R. Goviindan, “Impact of Network Density on Data Aggregation in Wireless Sensor Networks,” in *Technical Report 01-750, University of Southern California*, November 2001.

[8] K. W. Fan, S. Liu, and P. Sinha, “On the potential of Structure-free Data Aggregation in Senso r Networks,” in *To be appear in Proceedings of INFOCOM 2006*, April 2006.

[9] G. J. Pottie and W. J. Kaiser, “Wireless Integrated Network Sensors.”

[10] S. Lindsey, C. S. Raghavendra, and K. M. Sivalingam, “Data Gathering in Sensor Networks using the Energy*Delay Metric,” in *Proceedings 15th International Parallel and Distributed Processing Symposium*, April 2001, pp. 2001–2008.

[11] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data Gathering Algorithms in Sensor Networks Using Energy Metrics,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, September 2002, pp. 924–935.

[12] L. Zhao, X. Hong, and Q. Liang, “Energy-Efficient Self-Organization for Wireless Sensor Networks: A Fully Distributed Approach,” in *Proceedings of the 47th Annual IEEE Global Telecommunications Conference*, vol. 5, November 2004, pp. 2728–2732.

[13] B. J. Culpepper, L. Dung, and M. Moh, “Design and Analysis of Hybrid Indirect Transmissions (HIT) for Data Gathering in Wireless Micro Sensor Networks,” in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, January 2004, pp. 61–83.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, August 2000, pp. 56–67.

[15] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed Diffusion for Wireless Sensor Networking.”

[16] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, “Impact of Network Density on Data Aggregation in Wireless Sensor Networks,” in *Proceedings of 22nd International Conference on Distributed Computing Systems*, July 2002, pp. 457–458.

[17] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, “Building Efficient Wireless Sensor Networks with Low-Level Naming,” in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, October 2001, pp. 146–159.

[18] B. Krishnamachari, D. Estrin, and S. Wicker, “The Impact of Data Aggregation in Wireless Sensor Networks,” in *Proceedings. 22nd International Conference on Distributed Computing Systems Workshops*, July 2002, pp. 575–578.

[19] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks,” in *Proceedings of the 5th symposium on Operating systems design and implementation*, December 2002, pp. 131–146.

[20] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, “Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks,” in *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2004, pp. 49–58.

[21] M. Ding, X. Cheng, and G. Xue, “Aggregation Tree Construction in Sensor Networks.”

[22] A. Scaglione and S. D. Servetto, “On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks.”

[23] A. Scaglione, “Routing and Data Compression in Sensor Networks: Stochastic Models for Sensor Data that Guarantee Scalability.”

[24] R. Cristescu and M. Vetterli, “Power Efficient Gathering of Correlated Data: Optimization, NP-Completeness and Heuristics.”

[25] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “On Network Correlated Data Gathering,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, March 2004, pp. 2571–2582.

[26] S. Pattern, B. Krishnamachari, and R. Govindan, “The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, April 2004, pp. 28–35.

[27] “ExScal,” http://www.cast.cse.ohio-state.edu/exscal/.

[28] M. Zorzi and R. R. Rao, “Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance,” in *IEEE Transactions on Mobile Computing*, vol. 2, Oct. - Dec. 2003, pp. 349–365.

[29] ——, “Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance,” in *IEEE Transactions on Mobile Computing*, vol. 2, Oct. - Dec. 2003, pp. 337–348.

[30] S. Jain and S. R. Das, “Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks,” in *Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, June 2005, pp. 22–30.

[31] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” in *Proceedings of ACM MOBICOM 2000*, August 2000, pp. 243–254.

[32] Q. Fang, J. Gao, and L. J. Guibas, “Locating and Bypassing Routing Holes in Sensor Networks,” in *Proceedings of INFOCOM 2004*, vol. 4, March 2004, pp. 2458–2468.

[33] “Kansei Testbed,” https://cast.cse.ohio-state.edu/kansei/.

[34] “Stargate,” http://platformx.sourceforge.net/home.html.

[35] Crossbow, “Mica2,” http://www.xbow.com.

[36] “TinyOS,” http://www.tinyos.net.

[37] Chipcon, “CC1000,” http://www.chipcon.com.