Structure-Free Data Aggregation in Sensor Networks

Kai-Wei Fan, Sha Liu, and Prasun Sinha

Abstract—Data aggregation protocols can reduce the communication cost, thereby extending the lifetime of sensor networks. Prior works on data aggregation protocols have 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—Anycasting, data aggregation, sensor networks, structure-free.

1 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 raw data. To conserve energy for a longer network lifetime, it is critical for the network to support a high incidence of in-network data aggregation.

Optimal aggregation can be defined in terms of total energy consumption for transporting the collected information from sensors to the sink. Based on the topology of the network, the location of sources, and the aggregation function, an optimal aggregation structure can be constructed. Various centralized structured approaches [1], [2], [3], [4], [5], [6] 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 [7], [8], [9], [10]. 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 [7] 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 applications with amorphous events, such as biological hazard, chemical hazard, or fire detection, the 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 [6] are not practical for dynamic scenarios due to excessive communication overhead for centralized computation. 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 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 this paper, we explore the potential of structure-free data aggregation for event-based sensor networks. *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 partially structured approach with our structure-free approach may further improve performance, but it is beyond the scope of this paper.

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

The contributions of this paper are fourfold. First, we observe that packets need to be aggregated early on their

• *The authors are with the Department of Computer Science and Engineering, The Ohio State University, 395 Dreese Lab, 2015 Neil Ave., Columbus, OH 43210.*
E-mail: {fank, liusha, prasun}@cse.ohio-state.edu.

Manuscript received 2 Feb. 2006; revised 21 Sept. 2006; accepted 31 Oct. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0037-0206.
Digital Object Identifier no. 10.1109/TMC.2007.1011.

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). Second, 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. Third, we perform a detailed evaluation using simulations to establish the benefits of structure-free data aggregation. Fourth, we implement the DAA and RW approaches on TinyOS and evaluate them in an indoor network of 105 Mica2 motes.

The organization of the rest of the paper is as follows: Section 2 presents background and related work. Section 3 presents the DAA and RW protocols. The performance of the combined approach that uses both DAA and RW is modeled in Section 4. The performance evaluation of the protocols using simulations is presented in Section 5. Results from the indoor testbed confirming the benefits of the DAA and RW approaches are presented in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

In wireless sensor networks, the communication cost is often several orders of magnitude larger than the computation cost. Pottie and Kaiser [11] reported that the energy consumption for executing 3,000 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 network lifetime. Various protocols have been proposed to route packets for facilitating 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.

2.1 Cluster-Based Approaches

In [1], Heinzelman et al. propose the LEACH protocol to cluster sensor nodes and let cluster-heads aggregate data and communicate with the base-station directly using high transmission power. The cluster-heads are randomly elected in each round to distribute energy consumption among all nodes. LEACH-C [2] uses the base-station to broadcast cluster-head assignment to further spreading out the cluster-heads throughout the network. Based on LEACH, Zhao et al. [12] refine the cluster-head election algorithm that does not require the participation of the base-station and scatters cluster-heads more evenly across the network. However, it requires every node to broadcast at its highest transmission power at the setup stage of each round, which limits it ability to conserve energy.

Lindsey and Raghavendra propose PEGASIS [3], which organizes all the nodes in a chain and lets them play the role of head in turn to conserve more energy. Since there is only one head node and there are no simultaneous transmissions, latency is an issue in PEGASIS. To address this, two chain-based PEGASIS enhancements are proposed in [4] and [5]. In [4], Lindsey et al. propose a binary hierarchical approach for CDMA-capable sensor nodes and, in [5], they propose a chain-based three level approach that allows simultaneous transmission for non-CDMA-capable sensor nodes. Based on LEACH and PEGASIS, Culpepper et al. propose Hybrid Indirect Transmission (HIT) [13]. HIT still uses LEACH-like clusters, but allows multihop routes between cluster-heads and nonhead nodes.

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

2.2 Tree-Based Approaches

In [9], [10], Intanagonwiwat et al. propose a Greedy Incremental Tree (GIT) to establish an energy-efficient path based on Directed Diffusion [14], [15]. Krishnamachari et al. [16] compare three data-centric routing schemes, Center at Nearest Source (CNS), Shortest Path Tree (SPT), and a variation of (GIT) which establishes the route between the sink and the nearest source first, to illustrate the advantage of data aggregation. They observe that GIT performs the best in terms of average number of transmissions. In [17], [18], 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 a child cache to estimate 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. Ding et al. use shortest path tree with parent energy-awareness in [19], where the neighboring node with the shortest distance to the sink that has higher residual energy is chosen as the parent. All the above tree-based data aggregation routing protocols need a lot of message exchanges to construct and maintain the tree. Zhang and Cao propose Dynamic Convoy Tree-Based Collaboration (DCTC) in [7]. In [8], they further optimize the tree reconfiguration schemes. Essentially, DCTC tries to balance the tree in the monitoring region to reduce the energy consumption. But, it 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 are not desired when the data rate is high. Also, the performance of DCTC highly depends on the accuracy of mobility prediction algorithms.

Another class of tree-based data-centric routing protocols considers sensing information entropy in the routing metric [20], [21], [22], [23]. However, source coding or even its approximation is hard to deploy since the real distribution of collected data is hard to predict. In [22], [23], Cristescu et al. study explicit communication taking joint entropies among nodes into the routing metric and propose approximation algorithms such as Leaves Deletion approximation and Balanced SPT/TSP tree. But, these algorithms are centralized. They assume the global knowledge of the information entropy of each sensor node and the joint entropy of each pair, which makes such approaches nontrivial to implement in practice. Pattern et al. study the impact of spatial correlation on routing for some special cases in [24] and derive the optimal cluster size for these cases. Although authors use cluster structure, the basic tree-based routing is maintained instead of transmitting packets to the base-station in one hop.

3 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 to achieve optimal aggregation. For example, the transmissions should proceed from leaves to the root for a tree. Structured approaches, though suited for data gathering applications, have high overhead for event-based applications. The other extreme is to use opportunistic aggregation where packets are aggregated only if they happen to meet at a node at the same time. There is no overhead of structure construction; however, it may result in inefficient data aggregation. 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 a node at the same time to 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 their children before transmission. 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 changes, 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.

3.1 Spatial Convergence

In this section, we present the Data-Aware Anycast (DAA) protocol, which achieves the goals described above. The idea behind DAA is, instead of constructing a structure in advance for optimal aggregation, which is impossible without global knowledge of the network topology and traffic pattern, an independent set among sources is created. Nodes in the independent set act as aggregation points. The independent set is created distributedly and automatically while packets are forwarded to the sink; therefore, it reduces the maintenance overhead of structured approaches. To better describe the DAA protocol, we made the following assumptions:

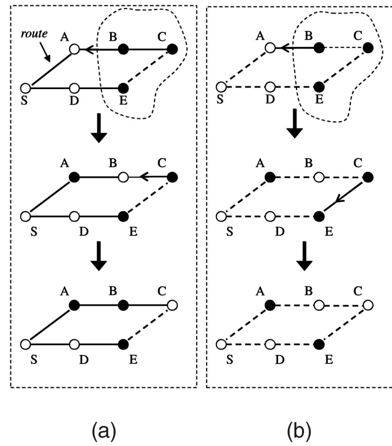


Fig. 1. Enhancing opportunistic aggregation with spatial convergence. (a) In opportunistic aggregation, nodes send their packet along the route constructed by the routing protocol. (b) Improving spatial convergence by allowing nodes to send packets to nodes that still have packets for aggregation. S 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. Nodes with data are represented with a dark circle.

- Nodes know the geographic location of their one-hop neighbors and the sink. Geographic information is essential in sensor networks and it can be acquired by GPS devices or localization protocols [25], [26].
- The interference range is at least twice the transmission range. This ensures that the neighbors of the sender will interfere with each other and no CTS from multiple nodes will collide. However, if this does not hold, other mechanisms, such as [27], can be used to prevent collision from multiple CTS packets.
- Nodes are time-synchronized. We aggregate packets that are generated at the same time; therefore, nodes have to be time-synchronized. However, if packets are aggregated according to other properties, such as geographic location, time-synchronization is not necessary.

When nodes send packets to the sink, they may follow different routes dictated by the routing protocol. Fig. 1 shows an example comparing opportunistic aggregation with optimal forwarding strategy. Fig. 1a shows the packet transmissions assuming opportunistic aggregation. Fig. 1b shows how information about the existence of data in neighboring nodes can be exploited to make dynamic forwarding decisions to achieve higher aggregation. The black circles 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 Fig. 1a. The distributed MAC protocol determines the order of transmissions in opportunistic aggregation, which does not achieve

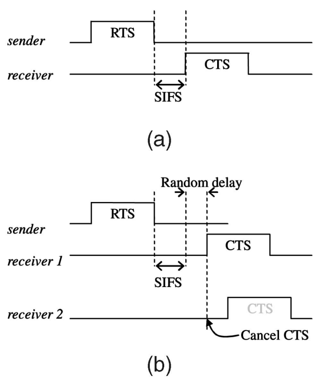


Fig. 2. Unicast versus anycast. (a) In 802.11, the receiver sends a CTS immediately after receiving the RTS. (b) In our approach, the receiver sends a CTS with a random delay to avoid collision between nodes sending the CTS. The CTS of receiver 2 has longer delay and, hence, is canceled after hearing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

any aggregation in this case. However, in Fig. 1b, if node C knows that node B does not have packets 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 three for opportunistic aggregation). This process can be repeated until a node does not have neighbors with packets for aggregation, such as E in Fig. 1b, and we call E 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 deployments of sensor networks, nodes have multiple choices for the next-hop. For example, in the ExScal [28] demonstration of the world's largest sensor network, each sensor had anywhere from three 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**. DAA is based on *anycasting* [27], [29], [30] at the MAC layer to determine the next-hop 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 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 can be any type of data, such as geographic location or time instance. In this paper, we use the measurement timestamp as the AID. Therefore, two packets that are generated 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 transmissions to avoid CTS collision. Fig. 2 shows the difference between unicasting in 802.11 and randomized CTS response in anycasting. Because we assume that the interference range is more than twice the transmission range, the neighbors of the sender can interfere with each other. Nodes will cancel their CTS transmission if they overhear any packet transmission during the random delay to prevent CTS collision.
- **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 packets to the sink using DAA, we enhance the mechanism as follows: Instead of dropping RTS if nodes do not have packets 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. 3a shows an example network of 50 nodes in a $200\ \mathrm{m} \times 200\ \mathrm{m}$ square with the sink at $(0, 0)$. The communication range of each node is $50\ \mathrm{m}$. Fig. 3b shows routes taken by packets using DAA 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 seven aggregation points (not including the sink). Compared to the 50 packets in the beginning, DAA reduces the packets to only seven without incurring any overhead of constructing or maintaining a structure.

Theoretically, the average number of aggregation points selected in DAA is $n/(k+1)$, where n is the number of nodes generating packets and k is the average degree of nodes.

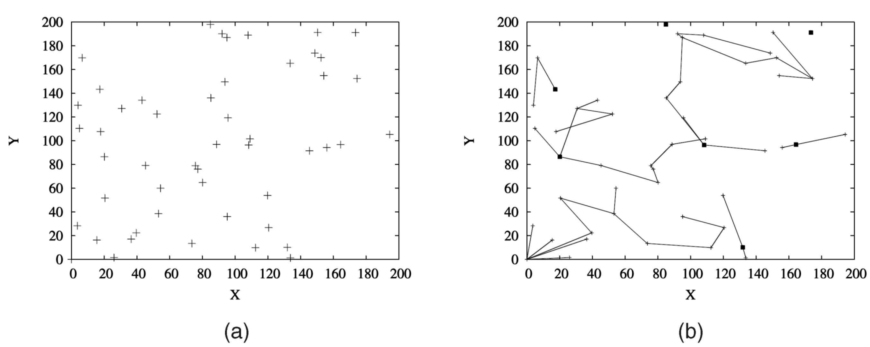


Fig. 3. (a) A 200×200 network with 50 nodes. (b) Routes constructed by the spatial convergence anycast protocol during anycast aggregation stage in the $200\ \mathrm{m} \times 200\ \mathrm{m}$ region with 50 nodes network. The sink is located at (0, 0) and the communication range of each node is 50 m.

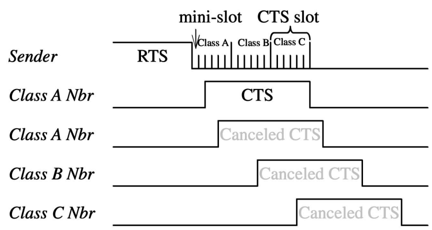


Fig. 4. Nodes with different priorities send CTS in different CTS slots, and nodes with the same priority select different minislots. A CTS is canceled if the node overhears another packet (or CTS).

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 remaining in the network is reduced by a factor of $k+1$ automatically, which saves 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.

3.1.1 CTS Priorities

Nodes are assigned different priorities in responding to an RTS. The three classes of priorities are:

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

Class B: The receiver has a 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 a 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 minislot 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 three 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 protocols indicating that 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 when there is no aggregation. Note that the actual transmission time of the CTS could be larger than the minislot or slot time. The slots and minislots are used to stagger the starting time of CTS transmissions. 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.

3.1.2 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. 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 the sender's location. The sender's location can be either contained in the RTS packet, 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 C can be restricted only to the designated next-hop on the perimeter. 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 (Goal 1). As there is no computed structure, event mobility has no impact on the performance of DAA (Goal 2). 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 (Goals 3 and 4). However, in the DAA approach, packets may not get aggregated if they are spatially separated (more than one 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.

3.2 Temporal Convergence

The second condition for aggregation requires packets to be present in 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 does not result in packets meeting temporally at intermediate nodes, the benefit of aggregation may be limited. The order of transmissions may be governed by several factors including interference from other flows and interference from the same flow.

Assume that the back off 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, nodes B and C will remain silent during the transmission. Therefore, no nodes will contend for the channel with A. Although C will send a CTS packet and the channel will not be idle for node A, node A will only back

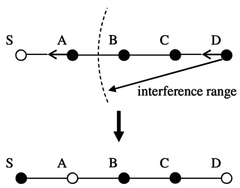


Fig. 5. A packet from D and A can hardly be aggregated if nodes forward the packet in the lock-step.

off 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 it 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.

Deterministically assigning the waiting time to nodes such that nodes closer to the sink wait longer can avoid the problem. However, nodes have no knowledge of the event size (the area in which nodes are triggered by the event) and location, and do not know their relative position compared to other nodes sensing the event. The only information that a node knows is its distance to the sink. Therefore, it can only set the delay inversely proportional to its 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 deployments.

Therefore, we propose Randomized Waiting (RW) at sources for each packet to introduce artificial delays and increase temporal convergence. Each source delays its transmission 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 and 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 nodes 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 size of the event and the time to transmit a packet. If the event size increases, i.e., 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 is too small, packets will not be aggregated even if 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 latency will be too high. If the application is not delay tolerant, a low value of τ is required which cannot reap the benefits of this approach. Since nodes are unable to know the size of the event, they cannot know the optimal value of the delay. However, due to the DAA's ability to achieve early aggregation even without delay, our approach is not very sensitive to the length of the delay. From the simulation results in Section 5, we learned that, by using RW together with DAA approach, the performance gain for delay longer than 1.6 seconds (about 40 packets transmission time) is marginal for events with 400 m in diameter. This suggests to us that a delay of few seconds is sufficient for the DAA+RW approach.

4 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.

4.1 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 upstream nodes, packets can be aggregated at downstream nodes. To simplify the analysis, nodes only forward packets to nodes closer to the sink, and we do not consider the transmission delay that may result in fewer aggregations.

Let Y be the discrete random variable representing the number of hops a packet has been forwarded when it is aggregated. Let X be the continuous random variable of the delay chosen by each sensor node, and its probability density and cumulative density functions be $f(x)$ and $F(x)$. Let $d_{v_h} = x$ be random delay 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 1) the packet is forwarded through $i-1$ hops and all nodes in these hops have lower delay than the sender and 2) at least one node at the ith hop has higher delay than the sender. Therefore, for a node that is h hops away from the sink,

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

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

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

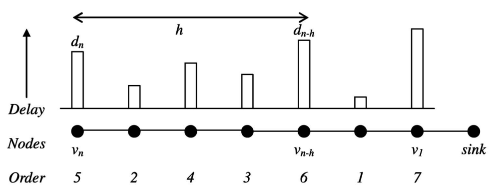


Fig. 6. The order of transmissions when nodes choose a random delay.

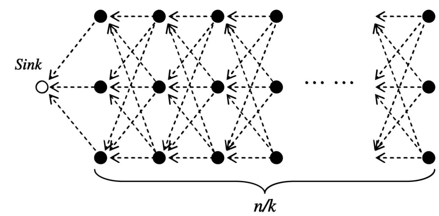


Fig. 7. A network topology where each node has $k = 3$ choices for downstream nodes.

Therefore, the expected value is

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

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 k downstream nodes, i.e., each level has k nodes, therefore the maximum hop number is n/k in average) and is

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

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

4.2 Alternate Analysis

From the above analysis, we know that the result is independent of the distribution of X. 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, where v_0 is the sink and all other nodes are sources. Picking a number from 0 to τ 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 and s_{v_n} is the second largest number. There are $\binom{n}{h+1}$ 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 $s_{v_{n-h}}$ and s_{v_n} are the largest two numbers. 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

$$\frac{\binom{n}{h+1} \times (h-1)! \times (n-(h+1))!}{n!} = \frac{1}{h(h+1)}.$$

If a packet of a node travels to the sink without any aggregation, the node must have highest delay and its probability is $\frac{1}{n}$. 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

$$E[Y] = \sum_{i=1}^{n} i \times P(Y = i) = \sum_{i=1}^{n-1} i \times \frac{1}{i(i+1)} + n \times \frac{1}{n} = \sum_{i=1}^{n} \frac{1}{i}. \tag{6}$$

The expected number of transmissions would be

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

We observe that this expression matches (4) for $k = 1$, validating the previous analysis.

4.3 Comparison with Simulation Results

To compare the analytical results with simulations (using *ns*2), 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 three downstream nodes in the next column. This corresponds to $k = 3$ in our analysis. We use $\tau = 2$ seconds for the maximum random delay for delaying packet transmission, which is an approximately 50 packet transmission time.

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 in 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 be aggregated at the downstream node with higher delay due to nonnegligible transmission delay. As this

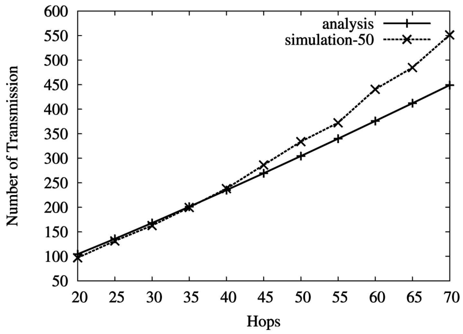


Fig. 8. The number of transmissions versus the network size for the analysis and simulation results.

effect increases with an increasing number of hops, the discrepancy increases accordingly. 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.

5 Performance Evaluation

In order to justify the design of the structure-free approach, we compare our protocol with a structured approach. We construct an aggregation tree rooted at the center of the event for each instance of measurement in advance. Nodes delay their transmission according to their height in the tree. Therefore, packets are transmitted from leaves to the root to achieve the highest number of aggregations. Parent nodes will transmit their packets once they receive packets from all their children, or when their delay timers expire. This approach should have the 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 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. The structure is a shortest path tree rooted at the sink, not rooted at the center of the event, as described above. Packets are aggregated only if they are at the same node at the same time (either at the application layer or MAC layer).
2. **Randomized Waiting (RW)**. Nodes send their packets along the shortest path to the sink with random delay at the sources. The RW approach falls back to the OP approach when the delay is 0.
3. **Data Aware Anycast (DAA)**. Nodes use spatial convergent anycast to aggregate packets without delaying at the source, as described in Section 3.1.
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, i.e., nodes with height 1 (leaves) do not delay, nodes with height 2 delay 0.64 seconds, etc., to get the best performance.

We use the *ns2* network simulator to evaluate these protocols. The RTS/CTS packet formats of 802.11 MAC are modified to incorporate the anycasting capability. The RTS packet 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. We assume that nodes know their neighbors' location.

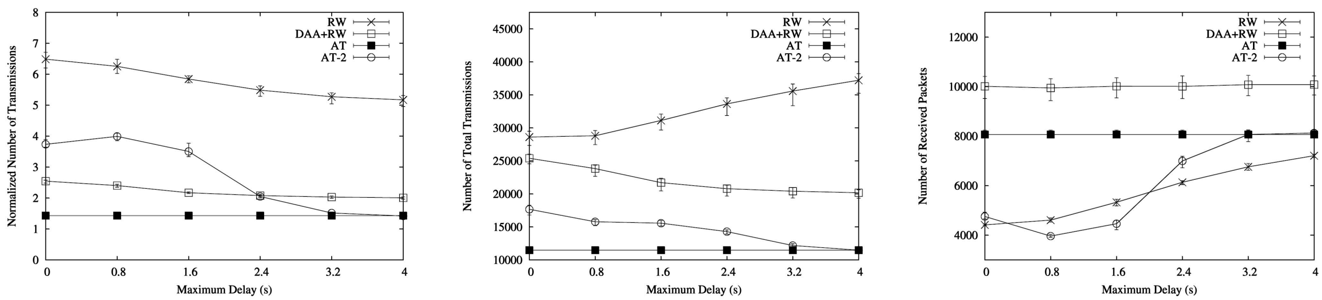
5.1 Simulation Scenario

The network is a $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ square region with grid topology. The sink is located at one corner of the network. The data rate of the radio is 38.4 Kbps and the communication range is slightly longer than 50 m. 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 200 m radius with 25 m as the distance between two nodes, there are 200 nodes generating packets at the same time (or 50 nodes generating packets if the distance between two nodes is 50 m). Unless otherwise mentioned, the internode separation is 30 m, the event moving speed is 10 m/s with a pause time of 0 seconds, the radius of the event is 200 m, and the maximum delay (τ) for RW and DAA+RW approaches is 3.2 seconds. All simulation results are based on five different mobility scenarios (each for 200 seconds). The minimum and maximum values obtained are also drawn in all graphs.

The normalized number of transmissions is used as the metric to compare different protocols. A 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 pieces of information that are generated by all sources in the network and are aggregated at the sink. The number of aggregations cannot tell if a protocol performs well since packets might be forwarded many hops before being aggregated. The number of transmissions will be lower if more packets are aggregated earlier. 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.

5.2 Maximum Delay

First, we evaluate the performance of the RW approach for achieving temporal convergence. Using the default scenario (30 m internode separation and 200 m event radius), there are around 140 sources and each source has about eight neighbors in its communication range. We vary the maximum delay from 0 (no delay) to 4 seconds, which is approximately the time to transmit 100 packets. Fig. 9 shows the results. In Fig. 9, AT uses 8 seconds as the maximum delay. If the height of the tree is higher than 13,



(a) (b) (c)

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

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 the AT approach with the maximum delay specified as the X-axis. OP and DAA are not shown in these graphs, but they are just RW and DAA+RW approaches with $\tau = 0$.

For protocols RW and DAA+RW, the normalized number of transmissions decreases as the maximum delay increases. At $\tau = 4$ seconds, the DAA+RW is 21 percent lower than DAA (DAA+RW with delay 0) and RW is 20 percent lower than OP. The structure approach AT, as predicted, has the lowest normalized number of transmissions. AT is 29 percent lower than DAA+RW in terms of normalized number of transmissions. However, AT requires the time and overhead of constructing the tree, which may degrade its performance further and is not shown in the graph. AT-2, the structure approach with different maximum delay, does not perform better than the DAA+RW approach when the maximum delay is less than 2.4 seconds. This shows that structure approaches are very sensitive to the waiting time as we described in Section 1. In the worst case, AT-2 has 300 percent more normalized transmissions than AT. However, with higher delay, the end-to-end delay is also higher for AT (Fig. 10). Because of DAA's ability to achieve early aggregation, it can effectively reduce the number of packets in the network even without delay. Therefore, DAA+RW is not as sensitive as the AT approach. Although, with higher values of τ, the normalized load of DAA+RW can be further reduced, beyond $\tau = 1.6s$, the reduction is marginal.

Figs. 9b and 9c show the number of transmissions and receptions of packets for different τ. We can see that AT has the lowest number of transmissions, which is only 57 percent of DAA+RW. However, DAA+RW has a higher number of received packets than AT. This is counterintuitive; since AT delays packet transmission nicely, it is expected to have a lower packet dropping rate. Tracing into the simulation logs, we found that the packet dropping rate is very high in AT. DAA+RW has about a seven packet loss, which counts for around 150 effective information, while AT has about a 200 packet loss, which counts for about 2,100 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 reduces the contention. Therefore, DAA and DAA+RW have lower packet loss and achieve a higher number of received packets.

We observe that, as DAA+RW achieves a lower number of transmissions than DAA and it decreases as τ increases, RW has opposite behavior to DAA+RW. RW has a higher number of transmissions than OP, and it increases as τ increases. This is because RW has a lower packet dropping rate than OP (in Fig. 9c). When τ increases, packets are more likely to be transmitted at different times, and the collision is reduced. As fewer packets are dropped, more packets are transmitted in the network, and they contribute to the increase in number of transmissions. The packet dropping rate is low for DAA as it achieves packet aggregation in the beginning, reducing the number of packets and lowering 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 containing three aggregated packets generated at time 0. At time 8, the sink receives another packet containing four aggregated packets 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 structured approaches have higher delay than the structure-free approaches. The main reason is that nodes have to wait for all their children before the

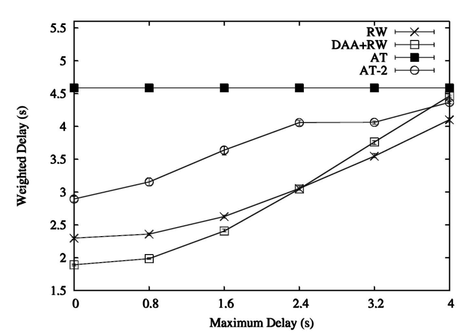


Fig. 10. The end-to-end transmission delay versus maximum delay.

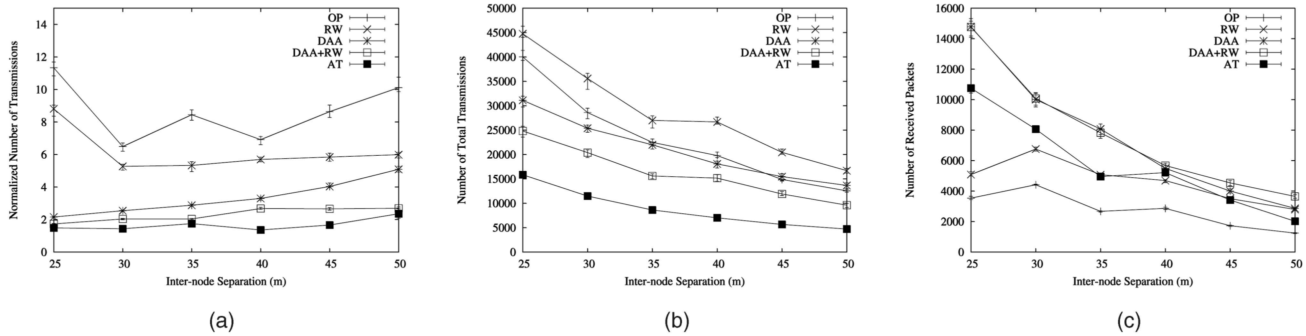


Fig. 11. The simulation results for different node densities. (a) Normalized number of transmissions. (b) Total number of transmissions in the network. (c) Number of received packets. The X-axis is the distance separating two adjacent nodes.

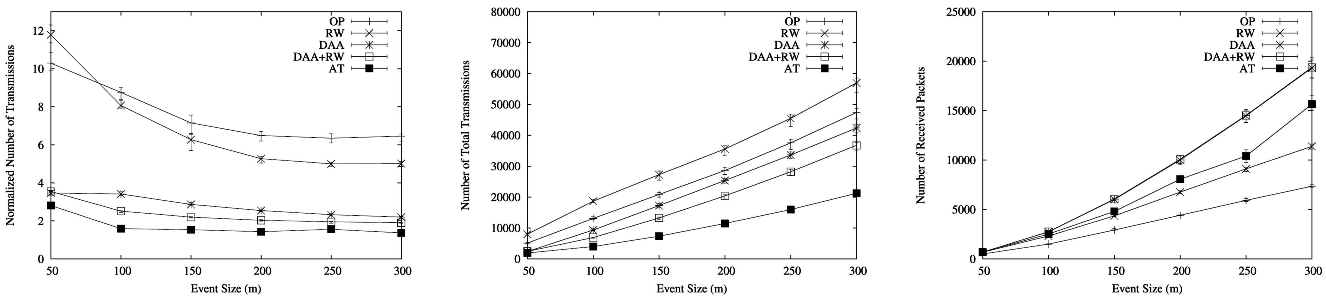


Fig. 12. The simulation results for different event moving speeds.

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 cannot achieve this since it is supposed to wait for packets from their children. For these applications, DAA+RW provides a better trade-off between energy and delay. For applications that are not tolerant to delay, DAA still provides good performance with lowest delay.

5.3 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. DAA and DAA+RW perform much better than the OP and RW approaches. At the highest node density, DAA+RW improves the normalized number of transmissions by 73 percent compared 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 percent lower than DAA+RW at the highest node density simulation (25 m internode separation).

DAA and DAA+RW have the highest number of received packets due to their ability to aggregate packets at the very beginning and scatter packets away from each other to reduce the contention and decrease the packet dropping rate. The results of OP and RW are not smooth because the packet dropping rate is very high. About 67 to 76 percent of packets are dropped during transmissions, compared with a less than 4 to 5 percent dropping rate for DAA and about 26 percent for AT. Because the total number of transmissions includes transmissions for 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.

5.4 Event Speed

Fig. 12 shows the results for different event moving speeds. We vary the event speed from 5 m/s to 20 m/s. We can see that the results remain steady at different speeds. As they do not create any structure for aggregation, mobility has little impact 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 overhead, the performance of AT will degrade as the speed increases.

5.5 Event Size

Fig. 13 shows the results for different event sizes. We vary the radius of the event size from 50 m to 300 m. The number of transmitted and received packets increases as the event size increases because more nodes are sensing the event. All protocols improve the performance as the event size increases. When the event size increases, more nodes are sending packets, and packets have more chances to be aggregated. Therefore, it reduces the normalized number of transmissions. With smaller event size, the performance of the OP and RW approaches degrades quickly, but the performance of the DAA and DAA+RW approaches only decreases slightly. This shows that DAA does get benefits from aggregating packets from close-by nodes; therefore,

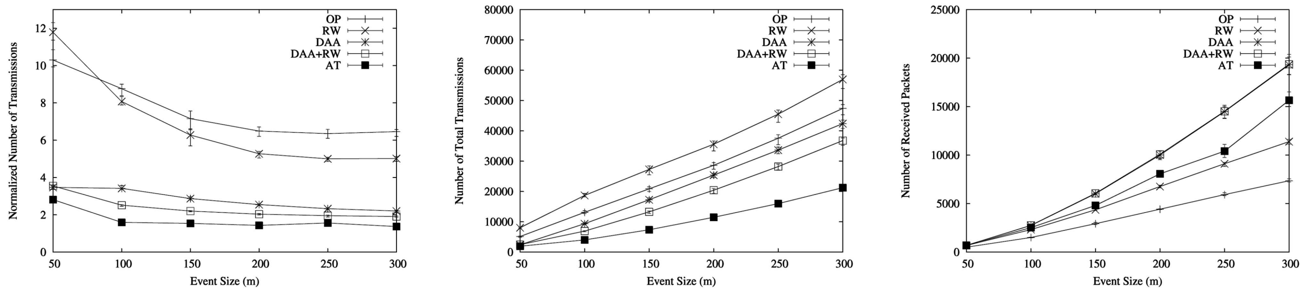


Fig. 13. The simulation results for different event size.

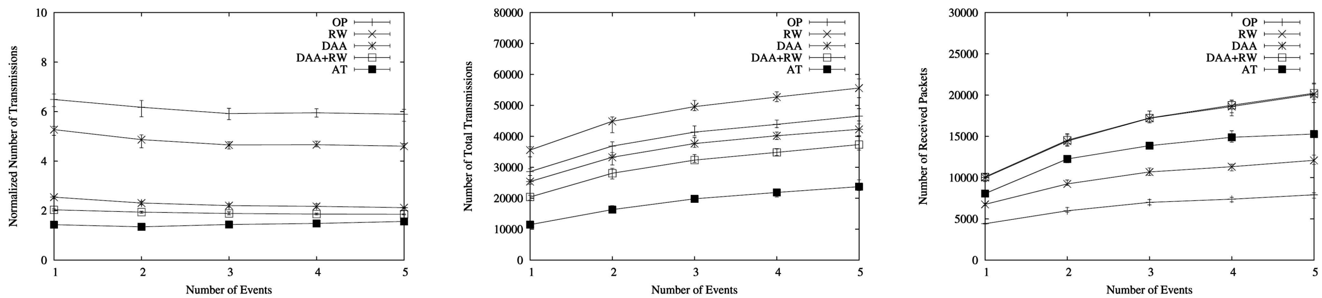


Fig. 14. The simulation results for different numbers of events.

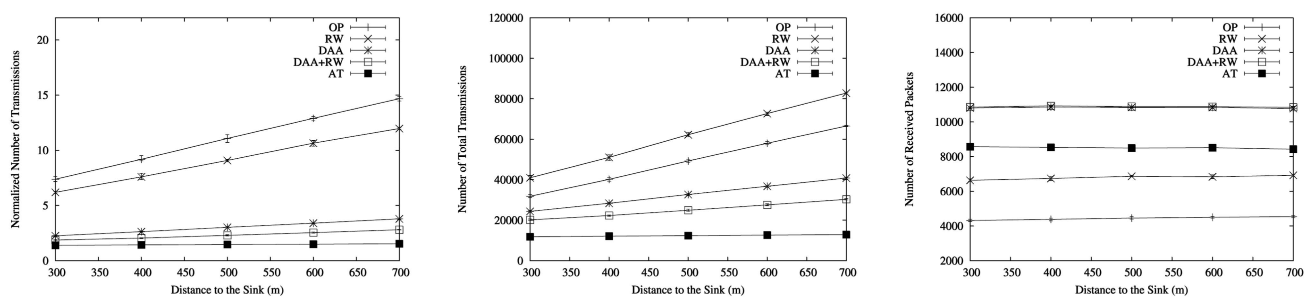


Fig. 15. The simulation results for varying distance between the event and the sink.

even though it is a structure-free approach, it still performs better than opportunistic aggregation.

5.6 Number of Events

Fig. 14 shows the results for different numbers of events in the network. The results are very similar to the experiment for event size. With a higher 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 compared to the increase in the number of events. This is because, within $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ networks with the 200 m radius of an event, most of the events are overlapped and the nodes only generate one packet no matter how many events are within its sensing range.

5.7 Distance to the Sink

Unlike AT, which can aggregate all packets into one node (optimally), 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 on average after the DAA approach, where n is the number of nodes generating packets and k is the average number of neighbors of a node. As more packets remain in the network, the cost of forwarding these packets is higher, and the increase is more significant when the event distance to the sink is longer.

We restrict an event to move only within a $200\ \mathrm{m} \times 200\ \mathrm{m}$ square centered at $(300, 300)$ to $(700, 700)$ and observe the performance of these 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 numbers of transmissions. Only AT remains the same across all scenarios. This suggests to us that semistructured approaches might be better at reducing the number of packets left in the network and improving the performance over structure-free approaches. This is considered as future work and is not in the scope of this paper.

5.8 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 the aggregation ratio for different protocols. When a node senses an event, it will generate a packet with 50 bytes of payload. We use a simple aggregation function to aggregate packets: The size of the

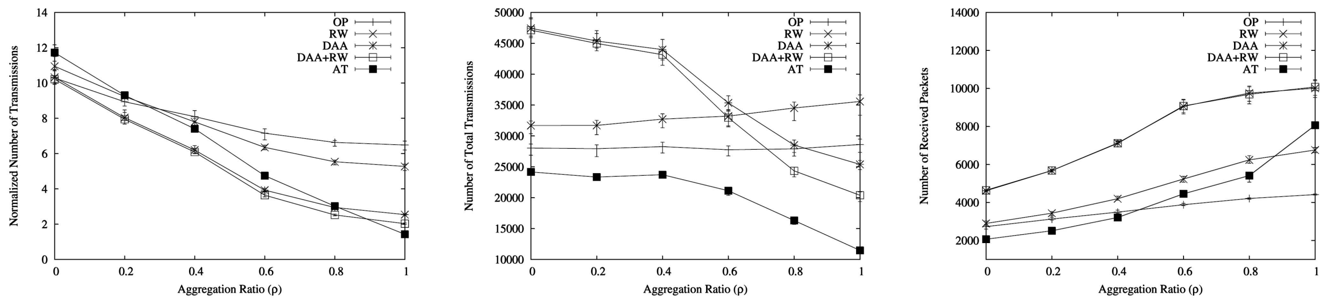


Fig. 16. The simulation results for different aggregation ratios. DAA and DAA+RW perform better than AT with aggregation ratio less than 0.8.

packet after aggregation is $max\{50, n \times (1 - \rho)\}$, where n is the amount 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 they receive an RTS from their neighbors. Therefore, we add a field to RTS to specify how many effective pieces of information are contained in the packet to let the node compute its priority for replying a CTS.

Fig. 16 shows the results for different aggregation ratios for different protocols. AT does not perform well for scenarios with aggregation ratios 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 cannot be aggregated anymore. In AT, when packets converge to the aggregation root, they become larger because they aggregate more packets. We observe that, when packets reach 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 contain more effective information. When they are dropped, more effective information is dropped.

In this simulation, we observe that structured approaches may not perform well if the aggregation is not perfect. By incurring the overhead of the structured approaches, their performance would be even worse. On the other hand, as the aggregation ratio decreases, AT cannot aggregate all packets 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 aggregate packets to one node. DAA and DAA+RW do 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 the structure approach in these scenarios.

6 Experiment Evaluation

We have implemented the DAA and RW approaches on the Kansei testbed [33], [34]. The Kansei testbed is composed of 210 stargates [35] arranged in a 15×14 grid. Each stargate is attached with a XSM/Mica2 mote [36] 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 [37]. It is equipped with multiple sensors and one CC1000 radio module [38]. This section introduces our experimental methodology and exhibits the advantage of using the DAA and RW approaches for improving the performance of data aggregation.

6.1 Experiment Design

The experimental topology is a 15×7 grid network with 105 XSMs. The XSM located at one corner of the network acts as the sink and the other 104 XSMs generate packets when they detect an "event." Each XSM can communicate with their two-hop grid neighbors directly, i.e., there are about 12 nodes within the transmission range. 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 sequence number (time-stamp in real deployed application) into a packet.

To mimic the event-triggering packet generation from XSMs at almost the same time, an event broadcast program runs on the Kansei server, which connects directly to all stargates through Ethernet. This program is also in charge of broadcasting statistics query messages to all stargates. Another program running on these stargates forwards the event message and statistics query message to the XSMs attached to these stargates, receives the statistics report message from the XSMs, and stores it in the disk.

We implement an anycast MAC protocol on top of the Mica2 MAC layer. The anycast MAC layer has its own back-off and retransmission mechanisms, and the ACK and back-off mechanisms of the Mica2 MAC module are disabled. The anycast MAC implements the RTS-CTS-DATA-ACK for anycast. Each XSM has a sending buffer for storing delayed packets, and both the application and MAC layer can access the sending buffer. When an XSM receives an event message, it generates a 16-byte Active Message containing current sequence number and puts it into the sending buffer. The XSM delays the transmission by starting a timer for a random period of time, and sends the packet to the MAC layer when the timer expires. If an

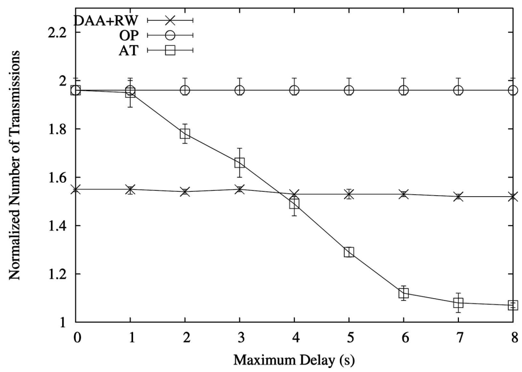


Fig. 17. Normalized number of transmissions on a 105-node grid network for DAA, OP, and AT approaches.

XSM receives an RTS, it checks both the MAC layer and the sending buffer for a packet with the same sequence number to decide its priority to reply with a CTS.

We compare DAA+RW to Aggregation Tree (AT) and Opportunistic Aggregation (OP), which are also implemented on TinyOS. Here, the AT approach creates a shortest path tree rooted at the sink (not rooted at the center of the event), which is constructed after the XSMs are deployed. In AT, nodes delay the transmission based on their height in the tree. Nodes closer to the sink have longer delay to wait for aggregating packets from their children. When a node receives a data packet, it checks both the MAC layer and the sending buffer if it has packet for aggregation. In OP, when a node receives the event message, it generates a packet and sends it to the MAC layer without any delay. When it receives a data packet, it only checks if its MAC layer has a packet for aggregation, and only aggregates a packet at the MAC layer.

6.2 Experiment Results

To compare the DAA+RW approach with the AT and OP approaches, the experiments are conducted for different maximum delays. We use a normalized number of transmissions to compare these approaches. Fig. 17 shows the experimental results of these protocols for different delays. When the delay is 0, the DAA+RW approach just falls back to DAA, and the AT approach just falls back to OP. DAA+RW outperforms the OP approach for all experiments. This shows that DAA+RW can efficiently aggregate packets and effectively reduce the number of packets in the network even with no preconstructed structure. DAA+RW performs better than AT when the maximum delay is less than 4 seconds. We can see that the performance of the DAA+RW approach improves slightly when the delay increases from 0 to 8 seconds. However, the AT approach is very sensitive to the delay. This indicates that the DAA+RW does not necessarily incur high end-to-end delay by using the RW approach.

7 Conclusion

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. We model the network load generated by the combined DAA with the 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 number of contributing sources (number of nodes whose packets reached the sink with or without aggregation). Using extensive simulations, we show that the combined DAA with RW approach can improve the normalized load by as much as 73 percent compared to opportunistic aggregation, and it performs better than the structured approach when the aggregation function is not perfect. Based on the experimental study with 105 XSM sensor nodes, we observe that the DAA+RW approach 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.

Acknowledgments

This material is based upon work supported by the US National Science Foundation under Grant No. CNS 0546630 (CAREER Award). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US National Science Foundation. The authors would like to thank Anish Arora and Rajiv Ramnath for allowing them to perform experiments on the Kansei testbed at Ohio State University.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Ann. Hawaii Int'l Conf. System Sciences*, vol. 2, Jan. 2000.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Comm.*, vol. 1, pp. 660-670, Oct. 2002.

[3] S. Lindsey and C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," *Proc. IEEE Aerospace Conf.*, vol. 3, pp. 1125-1130, Mar. 2002.

[4] S. Lindsey, C.S. Raghavendra, and K.M. Sivalingam, "Data Gathering in Sensor Networks Using the Energy*Delay Metric," *Proc. 15th Int'l Parallel and Distributed Processing Symp.*, pp. 2001-2008, Apr. 2001.

[5] S. Lindsey, C. Raghavendra, and K.M. Sivalingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 8, pp. 924-935, Sept. 2002.

[6] J. Wong, R. Jafari, and M. Potkonjak, "Gateway Placement for Latency and Energy Efficient Data Aggregation," *Proc. 29th Ann. IEEE Int'l Conf. Local Computer Networks*, pp. 490-497, Nov. 2004.

[7] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Trans. Wireless Comm.*, vol. 3, pp. 1689-1701, Sept. 2004.

[8] W. Zhang and G. Cao, "Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks," *Proc. INFOCOM*, vol. 4, pp. 2434-2445, Mar. 2004.

[9] C. Intanagonwiwat, D. Estrin, and R. Goviindan, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," Technical Report 01-750, Univ. of Southern California, Nov. 2001.

[10] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," *Proc. 22nd Int'l Conf. Distributed Computing Systems,* pp. 457-458, July 2002.

[11] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, pp. 51-58, May 2000.

[12] L. Zhao, X. Hong, and Q. Liang, "Energy-Efficient Self-Organization for Wireless Sensor Networks: A Fully Distributed Approach," *Proc. 47th Ann. IEEE Global Telecomm. Conf.,* vol. 5, no. 11, pp. 2728-2732, Nov. 2004.

[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," *ACM SIGMOBILE Mobile Computing and Comm. Rev.,* vol. 8, pp. 61-83, Jan. 2004.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. Sixth Ann. Int'l Conf. Mobile Computing and Networking,* pp. 56-67, Aug. 2000.

[15] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Trans. Networking,* vol. 1, pp. 2-16, Feb. 2003.

[16] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proc. 22nd Int'l Conf. Distributed Computing Systems Workshops,* pp. 575-578, July 2002.

[17] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," *Proc. Fifth Symp. Operating Systems Design and Implementation,* pp. 131-146, Dec. 2002.

[18] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler, "Supporting Aggregate Queries over Ad-Hoc Wireless Sensor Networks," *Proc. Fourth IEEE Workshop Mobile Computing Systems and Applications,* pp. 49-58, June 2004.

[19] M. Ding, X. Cheng, and G. Xue, "Aggregation Tree Construction in Sensor Networks," *Proc. 58th IEEE Vehicular Technology Conf.,* vol. 4, pp. 2168-2172, Oct. 2003.

[20] A. Scaglione and S.D. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," *Proc. Eighth Ann. Int'l Conf. Mobile Computing and Networking,* pp. 140-147, Sept. 2002.

[21] A. Scaglione, "Routing and Data Compression in Sensor Networks: Stochastic Models for Sensor Data that Guarantee Scalability," *Proc. IEEE Int'l Symp. Information Theory,* p. 174, June 2003.

[22] R. Cristescu and M. Vetterli, "Power Efficient Gathering of Correlated Data: Optimization, NP-Completeness and Heuristics," *Summaries of MobiHoc 2003 Posters,* vol. 7, pp. 31-32, July 2003.

[23] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On Network Correlated Data Gathering," *Proc. 23rd Ann. Joint Conf. IEEE Computer and Comm. Soc.,* vol. 4, pp. 2571-2582, Mar. 2004.

[24] S. Pattern, B. Krishnamachari, and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks," *Proc. Third Int'l Symp. Information Processing in Sensor Networks,* pp. 28-35, Apr. 2004.

[25] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm.,* vol. 7, Oct. 2000.

[26] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," *Proc. Second ACM Sensys Conf.,* pp. 50-61, 2004.

[27] S. Jain and S.R. Das, "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks," *Proc. Sixth IEEE Int'l Symp. World of Wireless Mobile and Multimedia Networks,* pp. 22-30 June 2005.

[28] "ExScal," http://www.cast.cse.ohio-state.edu/exscal/, 2007.

[29] M. Zorzi and R.R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Trans. Mobile Computing,* vol. 2, no. 4, pp. 349-365, Oct.-Dec. 2003.

[30] M. Zorzi and R.R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Trans. Mobile Computing,* vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.

[31] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. MobiCom,* pp. 243-254, Aug. 2000.

[32] Q. Fang, J. Gao, and L.J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," *Proc. INFOCOM,* vol. 4, pp. 2458-2468, Mar. 2004.

[33] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, "Kansei: A Testbed for Sensing at Scale," *Proc. Fourth Symp. Information Processing in Sensor Networks (IPSN/SPOTS Track),* 2006.

[34] A. Arora, E. Ertin, R. Ramnath, W. Leal, and M. Nesterenko, "Kansei: A High-Fidelity Sensing Testbed," *IEEE Internet Computing,* Mar. 2006.

[35] "Stargate," http://platformx.sourceforge.net/home.html, 2007.

[36] Crossbow, "Mica2," http://www.xbow.com, 2007.

[37] "TinyOS," http://www.tinyos.net, 2007.

[38] Chipcon, "CC1000," http://www.chipcon.com, 2007.

Kai-Wei Fan received the BS and MS degrees in computer science and information engineering from National Chiao Tung University, Taiwan, in 1997 and 1999, respectively. He was a senior engineer and project manager in the network security division of a start-up company from 1999 to 2004. He is currently pursuing the PhD degree in the Department of Computer Science and Engineering at Ohio State University. His research interests include wireless sensor networks and mesh networks where he focuses on the design and implementation of energy-efficient protocols.

Sha Liu received the BS degree in statistics and the MS degree in computer science from the University of Science and Technology of China in 2001 and 2004, respectively. Currently, he is a PhD student in computer science at Ohio State University. His research interests include low latency routing, wake-up scheduling, and data aggregation in wireless sensor networks.

Prasun Sinha received the PhD degree from University of Illinois, Urbana-Champaign (UIUC) in 2001, the MS degree from Michigan State University (MSU) in 1997, and the BTech degree from IIT Delhi in 1995. He worked at Bell Labs, Lucent Technologies, as a member of the technical staff from 2001 to 2003. Since 2003, he has been an assistant professor in the Department of Computer Science and Engineering at Ohio State University. His research focuses on the design of network protocols for sensor networks and mesh networks. He has served on the program committees of various conferences including INFOCOM (2004-2007) and MobiCom (2004-2005). He has won several awards, including the Ray Ozzie Fellowship (UIUC, 2000), Mavis Memorial Scholarship (UIUC, 1999), and Distinguished Academic Achievement Award (MSU, 1997). He received the prestigious US National Science Foundation CAREER award in 2006.