

Coverage Preserving Aggregation Protocols for Dense Sensor Networks

Jie Feng

Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5C9, Canada
Email: jif226@cs.usask.ca

Derek Eager

Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5C9, Canada
Email: eager@cs.usask.ca

Dwight Makaroff

Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5C9, Canada
Email: makaroff@cs.usask.ca

Abstract—Sensor networks are often deployed more densely than would be minimally required. In such cases, node scheduling protocols can be used to determine which nodes are active, and which nodes sleep so as to conserve energy and prolong network lifetime. A drawback of node scheduling approaches, however, is delay due to node or communication failure(s), and subsequent wake-up of replacement node(s), during which monitoring coverage of some sub-region may be lost. This paper proposes an alternative approach for use in contexts in which the objective is to periodically collect sensing data that completely covers a region of interest. In the proposed approach, nodes dynamically determine during each round of data collection whether they should transmit their data, or whether their area is covered by neighbouring nodes that have already transmitted. Both unicast and broadcast-based data collection protocols are designed, and their performance compared using simulation to that of data collection protocols relying on node scheduling. Our results suggest that the coverage-preserving broadcast-based protocol can greatly improve reliability at the potential cost of increased traffic volume owing to non-minimal selection of transmitting nodes.

I. INTRODUCTION

Area coverage is a common requirement in sensor network applications [1]–[3]. Sensor nodes must be deployed over some geographic region such that each point within the region is covered by the sensing capability of at least one node. We assume that sensors close to each other have correlated readings to be representative of the overlapped area. One approach to achieving area coverage is through optimized placement of a minimal number of sensors [4], [5]. Often, however, it is not feasible to optimize placement, and in any case, substantial redundancy may be desired owing to the possibility of sensor node and/or communication failures. For these reasons, sensor nodes may be deployed much more densely than would minimally be required.

To conserve energy and prolong network lifetime, node scheduling protocols may be used that cycle nodes between active and sleep states [1], [6]–[10]. The set of active nodes at any given point in time is sufficient to achieve the desired area coverage. A potential drawback of node scheduling approaches, however, is that there may be a significant delay

between (persistent or transient) node and/or communication failures, and subsequent wake-up of replacement node(s). During this time area coverage may not be complete. One way to address this problem is to choose the set of active nodes so as to provide redundant coverage, with each point covered by $k > 1$ active nodes [9]. There is a tradeoff here between reliability and energy use. Higher network load and contention are also possible with more nodes in the active state. Note that in failure-prone environments, even for $k = 2$ there may be significant probability of a loss of area coverage.

In this paper, we propose an alternative approach for efficiently maintaining area coverage in dense sensor networks. This approach is applicable in contexts in which the objective is to periodically collect, and transmit to a sink node, sensing data that covers the region of interest. In the proposed approach, rather than being (semi-)statically scheduled, nodes *dynamically* determine during each round of data collection whether they should transmit their data, or whether the set of neighbouring nodes that have already transmitted is sufficient to provide coverage.

In comparison to having all nodes collect and transmit their data, the proposed approach substantially reduces network traffic load and contention. Energy savings depend on the relative energy costs of data transmission, receiving data, idle listening, and sleeping. Techniques have been proposed that can substantially decrease the energy cost of the idle listening state [11].

In comparison to node scheduling approaches, the simulation results that we present in Section V suggest that the proposed approach can greatly improve reliability (yielding an order of magnitude reduction in uncovered area, in some cases), at the potential cost of increased traffic volume because the selection of transmitting nodes is not minimal. We design specific aggregation protocols implementing this approach and compare them with corresponding node-scheduling protocols [12]–[15]. For some of the protocols we consider, we assume the sink (and intermediate nodes) may receive multiple aggregates including the same sensor value, either because aggregation is duplicate insensitive (e.g., only the maximum sensor reading is needed), or duplicates can be filtered (each aggregate is a concatenation of sensor values).

In contrast to implementing a node scheduling protocol, and

To appear in Proc. LCN 2010, Denver, CO, Oct. 2010. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

separately an aggregation protocol that operates among the active nodes, we design protocols that integrate aggregation together with dynamic determination of which nodes transmit during each data collection round for both unicast-based and broadcast-based communication. In the unicast-based protocol, an aggregation tree is formed among all nodes, with the sink as the root [12], [16]. Each round, the interior tree nodes aggregate the data received from their child nodes, and forward their aggregate packets towards the sink. The leaf nodes rely on *overhearing* to determine whether or not their sensing area has been covered by neighboring nodes that have already transmitted during the round, and if so, refrain from transmitting. The key problem in the design of such a protocol is how to dynamically determine node transmission orderings that are “efficient”, i.e. nodes crucial to achieving area coverage transmit earlier and so enable other nodes that hear their transmissions to remain silent.

With the unicast-based protocol, failure of one of the interior tree nodes results in the loss of all of the data from the corresponding sub-tree. The broadcast-based protocol that we design addresses this weakness by eliminating the static tree structure, using instead a ring topology [17]. As with the unicast-based protocol, the key design problem is that of dynamically determining the transmission orderings in a coverage-aware manner.

We compare the new protocols with conventional unicast-based and broadcast-based data aggregation protocols relying on node scheduling, using simulation. Our comparisons are conservative with respect to the amount of generated network traffic in the node scheduling approach, since we assume optimal selection of a minimal coverage set. In practice, this would not be possible, owing at least to the need for cycling nodes between active and sleep states, rather than statically choosing a minimal coverage set to remain active at all times. The performance of the protocols is evaluated for both an independent random error model, and a two-state Gilbert error model. For both error models, the proposed broadcast-based protocol is found to provide greatly improved reliability in some cases, at the potential cost of increased traffic volume. The unicast-based protocol, in contrast, is found to yield similar reliability as unicast-based aggregation with node scheduling.

The remainder of the paper is organized as follows. Section II reviews related work. Section III presents our new unicast-based and broadcast-based coverage preserving aggregation protocols. Section IV describes the baseline protocols relying on node scheduling against which our performance comparisons will be made. Simulation results comparing the performance of the new protocols with the baseline protocols are given in Section V. Section VI concludes the paper.

II. RELATED WORK

There has been considerable prior work on node scheduling in densely deployed sensor networks, under some area coverage constraint. Both deterministic and statistical constraints have been considered. With deterministic constraints [1], [3],

[6], node scheduling must guarantee the desired coverage level whenever this coverage is possible to achieve. Note that this requires precise sensor node location information.

Examples of node scheduling protocols providing statistical area coverage are LDAS (Lightweight Deployment-Aware Scheduling) [7], PEAS (Probing Environment and Adaptive Sensing) [8], and RIS (Randomized Independent Scheduling) [9]. In LDAS, each node keeps track of the number of working neighbors. When this number exceeds a threshold computed based on the coverage requirement, the node randomly selects some of its working neighbours and sends tickets to them. When a node collects enough tickets, it goes to sleep after a random back-off time. The PEAS protocol attempts to minimize the state information kept at each sensor node. Nodes sleep for a randomized period of time, with mean value adaptively determined. When a node wakes up, it broadcasts a probe message that determines if there are any nearby active nodes. If there are, the node goes back to sleep. In RIS, at the beginning of each time cycle, each node decides to stay active with a probability p . For this scheme, Kumar *et al.* analyze how many sensors should be deployed in an area so that every point of the area is almost always covered by at least k sensors [9].

While node scheduling protocols aim to reduce or eliminate redundancy in the *set of active sensor nodes*, data aggregation aims to reduce redundancy in the *data traffic* that is generated by these active nodes. Our proposed protocols dynamically determine which nodes transmit in any particular round of data collection/aggregation. Thus the energy cost of the idle listening state is an important factor in the energy efficiency of these protocols. Jurdak *et al.* [11] quantify the energy required for unicast-based aggregation when static and semi-static node scheduling protocols are used, together RFIDImpulse. When using the wake-up radio protocol in RFIDImpulse, total energy usage is proportional to packets transmitted.

Existing protocols for data collection using aggregation may be classified as either unicast-based [12], [13], [16] or broadcast-based [17]–[20]. TAG is a unicast-based aggregation service [12] in which each node, beginning with the sink, informs its children in an aggregation tree of the interval during which it will be receiving data. A child’s transmission interval is fixed as the receiving interval of its parent. Thus, the sensors at the i -th level of the aggregation tree, $1 \leq i \leq H$, share transmission interval $H - i$, where H denotes the height of the tree. Gabriel *et al.* [18] and Motegi *et al.* [19], have designed broadcast-based aggregation protocol mechanisms for “duplicate sensitive” aggregation in which the sink must never receive multiple aggregates including the same sensor value. Our previous work introduced broadcast-based protocols for the case in which it is acceptable for nodes to receive multiple aggregates including the same sensor value [20].

III. COVERAGE PRESERVING AGGREGATION (CPA)

We assume a context in which sensor readings are made periodically with period duration τ . Sufficient data is to be returned to the sink each round, so that for each point in the

monitored region, the sink receives the data from at least one sensor whose sensing range covers that point.

The ‘‘coverage preserving’’ aggregation protocols that we propose integrate data collection using aggregation, together with dynamic determination of the nodes from which data should be collected. Each round, each non-sink node will transmit only if its sensing area is not completely covered by those neighbouring nodes that have already transmitted, and/or if the node must forward data received from other nodes. The goal is to minimize the number of transmitting nodes while ensuring no (or minimal) loss of area coverage.

We assume each node has a deterministic sensing range [1], [6]–[9] and knows its own location and the locations of its neighbouring nodes. A point p is covered by a node n if the distance between p and n is less than the node’s sensing range R ; i.e., node n provides coverage of a region bounded by a circle with radius R . It is assumed that the transmission range of each node is sufficiently large to reach all nodes whose sensing area overlaps with its own.

As in TAG [12], rounds are divided into intervals of identical duration I , and nodes that are h hops away from the sink are scheduled to transmit during interval $H - h$, where H is the maximum number of hops. We assume each node i knows its hop count h_i . For both unicast and broadcast protocols, all nodes agree on the same base time T_0 defining the beginning of the first round.

Each interval is divided into two phases, and each node transmits during one or both of the phases, depending on conditions explained in each following subsection. The relative length of each phase is determined adaptively by the relative traffic volume.

The percentage of the interval that is used for communication is bounded by the time needed as a gap, denoted by λ , between the two phases of transmission, and between two transmission intervals. The gap allows transmissions from one phase/interval to finish before the next phase/interval begins. Previous experiments have shown that $\lambda = 0.1I$ provides good end-to-end loss rates. The remainder of the interval is used for the two phases, in this case $0.8I$.

To spread out transmissions within the first phase of an interval, a random delay value $\Delta 1_i^j$ is chosen in each round j by each node i , such that the transmit time is $T_0 + (j - 1)\tau + (H - h_i)I + \Delta 1_i^j$ if the transmission is in phase 1. The range of $\Delta 1_i^j$ is $[0, (I - 2\lambda)f_i]$. The value of f_i is determined as described in Section III-A. If the transmission is in phase 2, another delay value $\Delta 2_i^j$ in the range $[0, (I - 2\lambda)(1 - f_i)]$ is chosen and the scheduled transmission time is $T_0 + (j - 1)\tau + f_i(H - h_i)I + (I - \lambda f_i) + \Delta 2_i^j$, as shown in Figure 1.

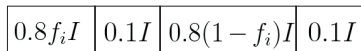


Fig. 1: Phase length determination

A. Unicast

The proposed unicast protocol is tree-based, with the union of the routes from the sensors to the sink forming an ag-

gregation tree with the sink as its root node, as shown in Fig. 2. Circles indicate sensing ranges, and it is assumed that the transmission range is twice the sensing range. H denotes the tree height.

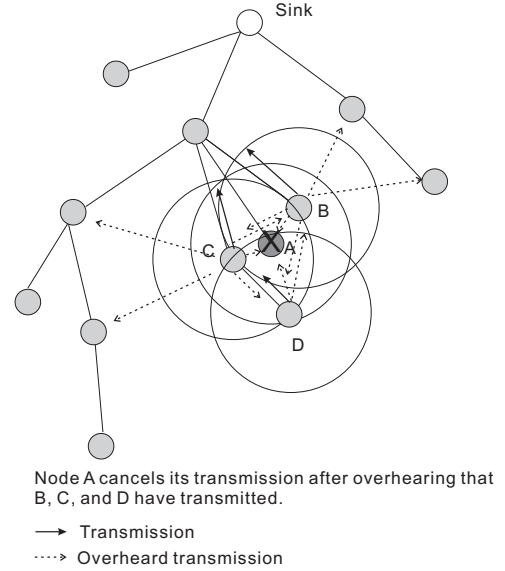


Fig. 2: Unicast CPA

In the initial round of data collection, $f_i = 1$ and there is no second phase. Each node i aggregates the data it has received for this round, and schedules its packet transmission according to the previously described Phase 1 formula. At the scheduled transmission time, node i actually transmits only if it is an interior (i.e., non-leaf) node, or if it has not overhead data transmissions, or acknowledgements of data transmissions, from a set of nodes that cover node i ’s sensing range.

For each subsequent round, there are two phases. Nodes that must forward data received from other nodes, and nodes that are dynamically determined to be important to achieving area coverage, schedule their transmissions for their first phase. In particular, since interior tree nodes must forward data received from their children, they transmit during their first phase.

The action taken by a leaf node depends in part on what happened during the previous round $j - 1$. Specifically, if, by the end of round $j - 1$, a leaf node i has overheard data transmissions, or acknowledgements of data transmissions, from a set of nodes that cover node i ’s sensing range, then node i schedules its transmission for round j to be during its second phase. Otherwise, node i schedules its transmission to be during its first phase. Again, node i actually transmits only if it has not overheard, during the current round j , data transmissions, or acknowledgements of data transmissions, from a set of nodes that cover node i ’s sensing range.

The relative durations of the phases into which node i divides its respective interval (as determined by the value of f_i) are dynamically determined so as to match the anticipated relative traffic volumes in node i ’s neighbourhood, using an Exponentially Weighted Moving Average (EWMA) strategy. Let F_i^j and S_i^j denote the number of first-phase and second-

phase data transmissions that node i hears from the nodes in the same ring during round. Each data packet includes a bit indicating which phase the transmitting node was in when the transmission occurred. At the end of round j , node i takes as its updated value of f_i a weighted average of the old value of f_i , and the measured fraction of first-phase transmissions during that round: $(1-\delta)f_i + \delta(F_i^j / (F_i^j + S_i))$, where δ (0.125 for the simulations of Section V) is a parameter determining the weight given to the most recent measurement.

B. Broadcast

The broadcast CPA protocol that we design organizes nodes into a ring, rather than tree, topology [17]. As shown in Fig. 3, The sink is the only node that is located in ring 0, nodes one hop away from the sink are in ring 1, and in general nodes h hops away are in ring h . As in the unicast protocol, nodes in different rings are allotted different time intervals within each round of communication for their transmissions. Denoting the maximum hop count from the sink by H , the nodes in ring h transmit in interval $H - h$. Each ring h node aggregates all of the data it has received from broadcasts for the current round (from neighbouring nodes in ring $h+1$, as well as from neighbouring nodes in the same ring h that transmitted earlier within the interval), for their own broadcasts. Each broadcast packet includes a bit vector indicating the nodes whose data is aggregated in the packet. It is acceptable for the sink to receive multiple aggregates including the same sensor value.

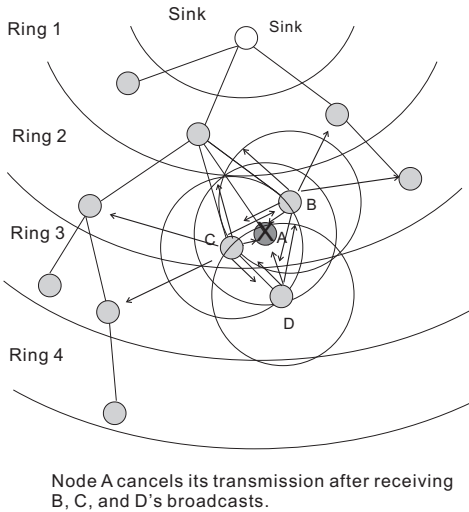


Fig. 3: Broadcast CPA

Each node i divides its respective interval into two phases, with gaps as previously mentioned, according to $\lambda = 0.1I$. For the first round, f_i is chosen as 0.9. For subsequent rounds, the value of f_i is updated as in the unicast protocol. To improve reliability, in some cases a node may transmit in both phases.

A scheduled transmission by a node i is cancelled if *both* of the following conditions hold at the time the transmission was to have occurred:

- for each node in ring $h_i + 1$ that node i has heard a broadcast from, for the current round, node i has also

heard a broadcast from some other node in ring h_i that has included that node's sensor data in its aggregate;

- the set of other nodes whose data node i has aggregated for the current round (from the broadcasts it has received) cover node i 's sensing area.

The first of these conditions is needed to ensure that for each node in a ring $h + 1$, there is at least one node in ring h that forwards its data.

For the first round, each node computes a time to transmit, based on the first phase formula. If this first broadcast is made, i.e., one or both of the above conditions did not hold, a second broadcast is scheduled (for improved reliability) within the second phase. This second broadcast is made only if one or both of the above conditions does not hold, *and* node i has not heard a broadcast from some other ring h_i node, subsequent to node i 's first phase broadcast, that includes its data from this broadcast.

For each subsequent round, nodes schedule transmissions much as described above for the first round, except that a transmission is scheduled for the first phase of the respective interval *only if* one or both of the above bulleted conditions did not hold at the end of the previous round. Otherwise, a transmission is scheduled only for the second phase.

IV. DATA COLLECTION WITH NODE SCHEDULING

The performance of the proposed protocols is compared to that obtained by using node scheduling together with conventional unicast and broadcast aggregation protocols. Our evaluation of this approach is optimistic with respect to the amount of generated network traffic. Rather than evaluate any particular practical node scheduling protocol, we apply an optimization algorithm to find a minimal coverage set (i.e., a smallest set of active nodes such that the sensing area they cover is the same as that of the set of all nodes). Data collection using aggregation is then performed over the nodes in this minimal coverage set (hereafter denoted MCSA). We use the conventional synchronous unicast and broadcast aggregation protocols [20] for comparison, and their deployment in this context is described in Sections IV-A and IV-B respectively.

A. Unicast Aggregation

The unicast aggregation protocol we consider matches the unicast CPA protocol proposed in III-A, with the following differences. An aggregation tree is used, but now just including the nodes in the minimal coverage set. Nodes at different levels of the tree are assigned to different intervals within each round of communication, as previously. Each round, nodes schedule their transmissions in their respective intervals exactly as is done in the first round in the CPA protocol. Unlike in that protocol, there is no protocol mechanism for cancelling a scheduled transmission.

B. Broadcast-based Aggregation

As in unicast aggregation, the broadcast-based MCSA protocol matches the broadcast CPA described in III-B, with the following exceptions. Each interval is divided into a first and

second phase, but unlike in CPA protocol, the durations of these phases are determined by a fixed parameter α (0.8 in the experiments). All nodes in the minimal coverage set transmit in the first phase, and some nodes also transmit in the second phase as well, for improved reliability.

Specifically, in each round j , node i picks a random value $\Delta 1_i^j$ between 0 and $(\alpha - 0.1)I$, aggregates the data from the broadcasts it has received for this round, and transmits at time $T_0 + t(j - 1) + (H - h_i)I + \Delta 1_i^j$. Node i then picks another random value $\Delta 2_i^j$ between 0 and $(1 - \alpha - 0.1)I$. A broadcast is made in the second phase, at time $T_0 + t(j - 1) + (H - h_i)I + \alpha I + \Delta 2_i^j$, if, by this time, node i has not heard a broadcast transmission from some other node in the same ring that has included node i 's data owing to the other node having heard node i 's first broadcast.

V. COMPARATIVE EVALUATION

Performance is evaluated using ns2 simulations. The metrics considered are (1) the average percentage of uncovered area in each round of data collection, (2) the average number of packet transmissions/receptions per round (for unicast, including both initial data packet transmissions, and link layer retransmissions and acknowledgements), (3) the average number of nodes that transmit per round, and (4) the average number of bytes transmitted/received per round. The last three metrics yield insight into relative energy usage. The uncovered area in each round is that area which is not covered by the nodes whose data is included in the aggregates that are successfully received at the sink. Node area coverage is determined based on a division of the region into cells. A cell is considered covered when the center of the cell is covered by a node.

Sensor fields are generated by randomly scattering nodes in square areas. The node closest to the center of the area is selected as the sink.¹ Sections V-A and V-B show results for a sensor network with 320 nodes deployed over a 250 metre by 250 metre area. The cell size we use to determine node area coverage is 2 metres by 2 metres, giving 15625 cells in total for this network. With all 320 nodes, about 99.4% of the area is covered. The size of a minimal coverage set covering the same area is 96 nodes. Section V-C shows performance results for both lower and higher density networks.

A MAC layer using CSMA/CA without RTS/CTS [21] is simulated for the unicast protocols, This MAC layer specification matches closely with the MicaZ motes² used in previous implementation experiments [20]. For simplicity, each node has a transmission range of 40 metres and maximum rate of 2 Mbps. Different maximum numbers of link layer retransmissions (3 and 8) are simulated for when the sender fails to receive an acknowledgement. The same transmission range and data rate are used for the broadcast-based protocols. All nodes have a sensing range of 20 metres.

The period duration τ (and corresponding interval duration I , chosen as τ divided by the maximum hop count to the

sink) is chosen sufficiently large that network contention is minimal. We make no assumptions regarding the type of aggregation that is performed, but consider two extreme cases with respect to how packet size grows with the number of aggregated values. In one of these, it is assumed that sensor data can be aggregated into packets of size that is independent of the number of aggregated values (chosen as 52 bytes in the simulations reported here). In the other, required packet size is assumed to increase linearly with the number of aggregated values (at 4 bytes per value).

Section V-A evaluates protocol performance under an independent random error model in which physical layer packet loss occurs independently for each packet, with a fixed probability, and in which there are no node failures. Section V-B considers performance with a two-state Gilbert error model that can reflect longer link outages and partial node failures, depending on the parameter settings. The impact of network density on performance is explored in Section V-C.

A. Performance for Independent Random Error Model

Figs. 4 to 6 show the performance of both the proposed CPA protocols, and the MCSA protocols, for the independent random error model. Each figure plots one of the performance metrics as a function of the physical layer packet loss rate (expressed as a loss probability).

Fig. 4 shows the average percentage of uncovered area in each round, when the packet size is fixed. Results for when the

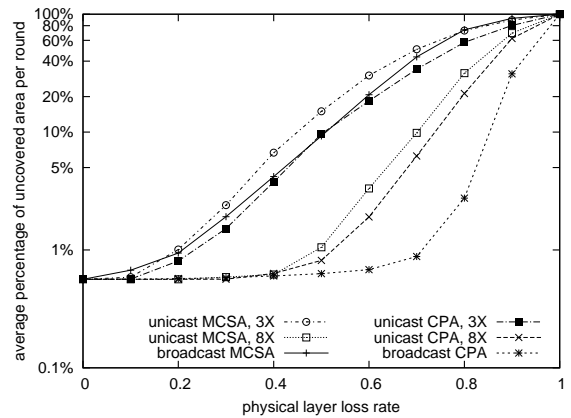


Fig. 4: Uncovered Area per Round: Independent Random Error Model (fixed packet size)

packet size increases with the number of aggregated values are similar. As the loss probability increases beyond 0.1, broadcast CPA significantly outperforms broadcast MCSA, and all of the unicast protocols with a link layer retransmission limit of 3. Broadcast CPA begins to outperform the unicast protocols with a link layer retransmission limit of 8, as well, once the loss probability exceeds 0.4. These results are explained by the fact that as the packet loss rate increases, the broadcast CPA protocol correspondingly increases the number of nodes transmitting their sensor data in each round, as seen in Fig. 5. The number of transmitting nodes is fixed at 96 for the minimal coverage set protocols.

¹When evaluating aggregation using a minimal coverage set, we (optimistically for this approach) select from only these nodes.

²Crossbow: <http://www.xbow.com>

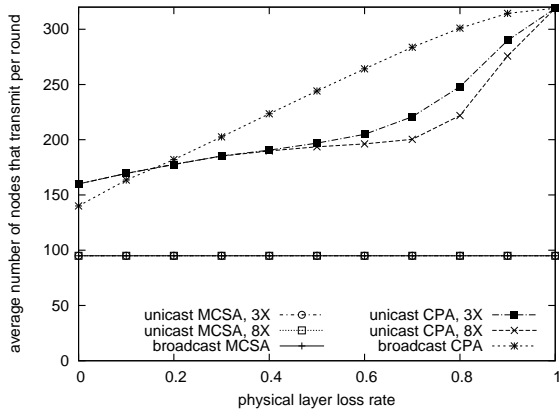


Fig. 5: Average Number of Nodes that Transmit per Round: Independent Random Error Model (fixed packet size)

Unicast CPA however, yields only modestly improved coverage in comparison to unicast MCSA. The relatively poor performance in comparison to that with broadcast CPA shows the impact of packet loss by the interior tree nodes.

The average number of bytes transmitted per round is shown in Fig. 6 for both the fixed and increasing packet size cases. Plots for the average number of packet transmissions per round are very similar to those in Fig. 6a, and are omitted.

For unicast, bytes representing link layer retransmissions and acknowledgments are included. For fixed packet size, the broadcast protocols send relatively few bytes; this is since each node transmits its data at most twice with these protocols, and since nodes do not send acks. Note that more bytes are sent with CPA than when using the corresponding MCSA protocol, even for a packet loss rate of 0. This reflects the fact that a greater number of nodes transmit with this approach, as seen in Fig. 5. For increasing packet size, the data volume with broadcast CPA is typically considerably larger than with the other protocols, owing to cases in which the same sensor value is redundantly included in multiple aggregates. The data volume with this protocol substantially decreases as the loss probability exceeds 0.7, since the growing packet loss decreases this redundancy.

Table I shows the traffic per round for different aggregation protocols, including conventional unicast and broadcast-based aggregation over all 320 nodes inside the network. Bytes transmitted/received per round are shown for the case of increasing packet size only. The corresponding results for fixed packet size can be easily computed from the number of packets transmitted/received per round. Compared with broadcast-based conventional aggregation with all nodes in the network, broadcast CPA significantly reduces the amount of transmitted/received data for both fixed and increasing packet size. While conventional unicast aggregation, and the MCSA protocols, have a lower amount of received data, they offer lower reliability than the broadcast CPA protocol.

For conventional unicast-based aggregation, we assume a node is woken up only when some other node wants to transmit to it, as in the RFIDImpulse protocol [11]. The node

Protocol	Transmissions data+ACK pkts	Receptions data+ACK pkts	Bytes transmitted	Bytes received
Unicast MCSA	95+95	95+95	9506	9506
Unicast CPA	160+160	2463+3100	15,796	261,672
Unicast all nodes	319+319	319+319	31,622	31,622
Broadcast MCSA	126+0	400+0	14,768	43,437
Broadcast CPA	178+0	2906+0	29,444	461,880
Broadcast all nodes	351+0	4020+0	136,136	1,330,011

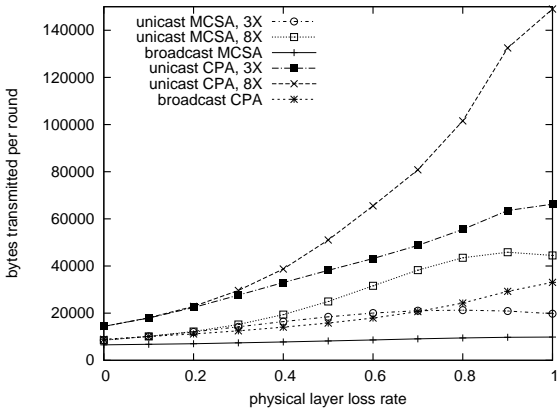
TABLE I: Traffic Per Round (increasing packet size, 0% physical layer loss rate)

goes back to sleep once the transmission is over. For the other protocols, the node in ring h has to wake up and listen to any transmissions it hears during a certain period of time. For a node in ring h , that period starts from the beginning of interval $H-h$ or $H-h-1$ for $h = H$ and $h < H$ respectively. The end of that period differs among the protocols. For conventional broadcast-based aggregation, the period ends when its own transmission interval ends, or when it hears that its current data has been rebroadcasted by some node, whichever comes first. For unicast and broadcast CPA, the period ends when its own transmission interval ends, due to the need to receive packets for the computation of f_i . With wake-up radio, the cost of idling listening can be minimized. Energy consumption is dominated by data transmissions and receptions.

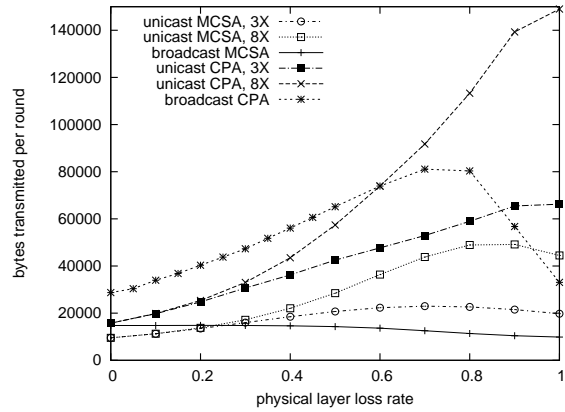
B. Performance for Two-state Gilbert Error Model

Figs. 7 and 8 show performance results for a two-state Gilbert error model. At each point in time, each node (other than the sink) is in one of two states, independently of all other nodes. In the “good” state, all packets are received correctly. In the “bad” state, no packets are received correctly. The time a node spends in a state before transiting to the other state (the sojourn time) is exponentially distributed. By varying the average sojourn time, a range of scenarios can be modelled, from independent random packet loss (with very short sojourn times) to link outages and partial node failures (with long sojourn times).

In Fig. 7, the proportion of time each node spends in the bad state is fixed at 20%, and the average sojourn time in that state (and the corresponding average sojourn time in the good state) is varied. Broadcast CPA provides better coverage for long duration failures than the unicast protocols, because the latter protocols rely on link layer retransmissions, which are ineffective in this case. Broadcast CPA also yields better coverage than broadcast MCSA, since the former protocol can often rely on other nodes to compensate for the nodes in the bad state. Note that for a very small average sojourn time, such as 0.0001 seconds, the average percentage of uncovered area in each round increases, since a node is likely to enter the bad state at least once while receiving a packet, causing the loss of that packet.



(a) Fixed Packet Size



(b) Increasing Packet Size

Fig. 6: Average Bytes Transmitted per Round: Independent Random Error Model

Qualitatively similar results are obtained for 33% of the time in the bad state. The reliability advantage of broadcast CPA, compared to the other protocols, increases as the reliability of the network decreases. Further tests were performed with 10% of time in the bad state showing similar effects. Fig. 8 shows the average number of bytes transmitted per round, for both the fixed and increasing packet size cases. Plots for the average number of packet transmissions per round are very similar to those in Fig. 8a, and are omitted. As is the case for the independent random error model, broadcast CPA yields a higher number of bytes transmitted. However, its relative performance with respect to this metric generally improves as the average state sojourn time in the Gilbert model increases.

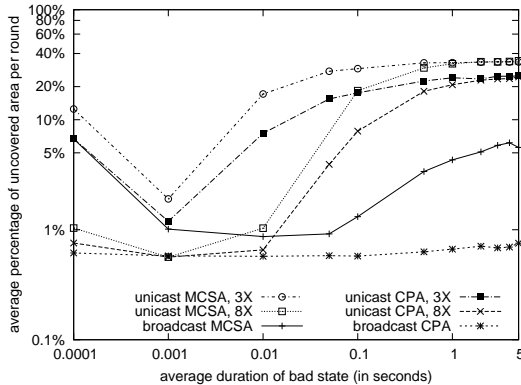


Fig. 7: Uncovered Area per Round: Gilbert Error Model (20% of time in bad state, fixed packet size)

C. Impact of Network Density

We performed experiments with different network densities, changing the area to 300 metres by 300 metres and 200 metres by 200 metres, respectively. Fig. 9 shows the results for higher density. The results for the lower density case are similar in shape to the results shown (Figs. 4 and 7), but indicate a smaller improvement with broadcast CPA. The main observation is the dramatic impact of density on the relative

performance of the protocols. In particular, higher density increases the reliability improvements with broadcast CPA.

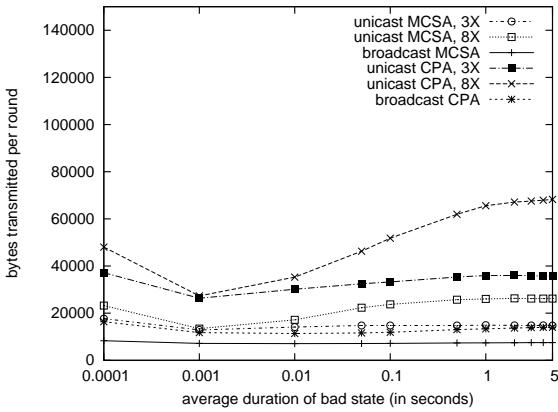
VI. CONCLUSIONS

In this paper, we proposed new protocols for data collection in dense sensor networks. The goal of these protocols is to maintain complete area coverage at a low cost with respect to the required sensor node transmissions. Our new protocols integrate aggregation together with dynamic selection of which nodes should transmit during each data collection round. Simulation results show that our new broadcast CPA protocol, in particular, is able to achieve substantially improved reliability in some cases (in comparison to node scheduling approaches in which only a minimal coverage set of nodes is active), at substantially lower cost than if all nodes transmitted during each round.

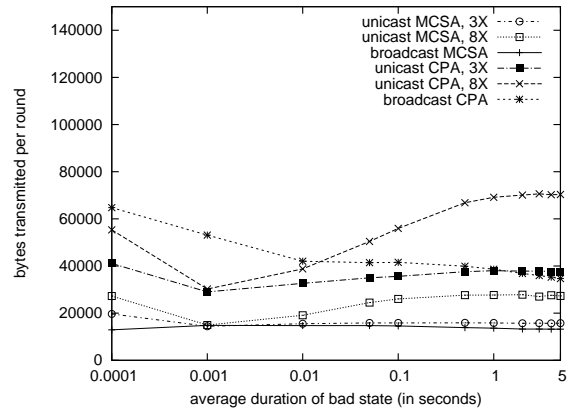
Our protocols use observations of transmissions from the previous round, to determine whether a node's transmission will likely be needed in the current round. If so, the transmission is scheduled early so that other nodes may hear it and possibly realize that their own transmissions are not needed. In general, this strategy results in a larger set of nodes transmitting in each round than is minimally necessary. A possible area of future work concerns hybrid strategies that use dynamic node selection, as in the proposed protocols, but that use additional topology information to assist in deciding which nodes may be most important to achieving area coverage.

REFERENCES

- [1] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *WSNA '02*, Atlanta, GA, Sept. 2002, pp. 32–41.
- [2] J. Carle, A. Gallais, and D. Simplot-Ryl, "Preserving Area Coverage in Wireless Sensor Networks by Using Surface Coverage Relay Dominating Sets," in *ISCC '05*, Washington, DC, June 2005, pp. 347–352.
- [3] H. Zhang and J. C. Huo, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc and Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 89–123, 2005.
- [4] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *IEEE WCNC '03*, New Orleans, LA, Mar. 2003, pp. 1609–1614.

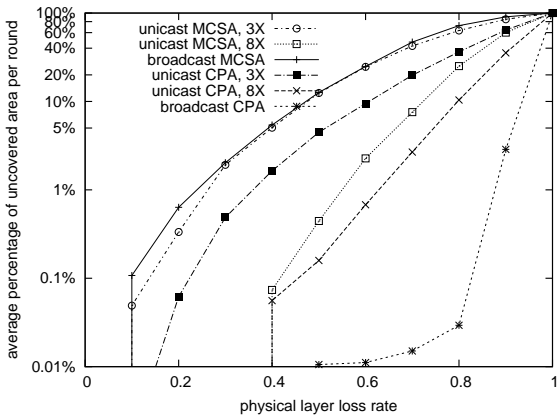


(a) Fixed Packet Size

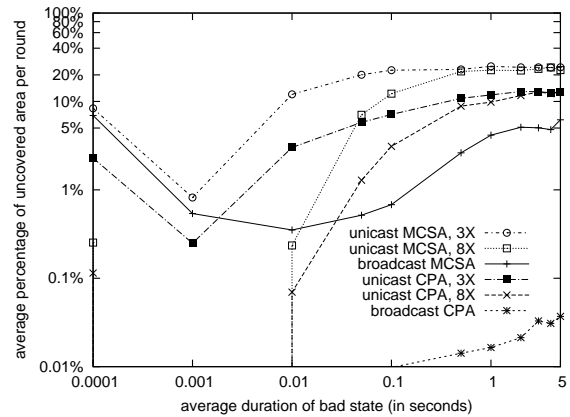


(b) Increasing Packet Size

Fig. 8: Average Bytes Transmitted per Round: Two-state Gilbert Error Model (20% of time in bad state)



(a) Independent Random Error Model



(b) Two-state Gilbert Error Model (20% of time in bad state)

Fig. 9: Performance for Higher Density Network (fixed packet size)

- [5] S. Zaidi, M. Hafeez, S. Khayam, D. McLernon, M. Ghogho, and E. Kim, "On minimum cost coverage in wireless sensor networks," in *Information Sciences and Systems '09*, Baltimore, MD, Mar. 2009, pp. 213–218.
- [6] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *SenSys '03*, Los Angeles, CA, Nov. 2003, pp. 28–39.
- [7] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 6, pp. 837–852, 2005.
- [8] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," in *ICDCS '03*, Providence, RI, May 2003, pp. 28–37.
- [9] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *MobiCom '04*, Philadelphia, PA, Sept.–Oct. 2004, pp. 144–158.
- [10] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *MobiCom '04*, Philadelphia, PA, Sept.–Oct. 2004, pp. 129–143.
- [11] R. Jurdak, A. G. Ruzzelli, and G. M. O'Hare, "Multi-hop RFID wake-up radio: Design, evaluation and energy tradeoffs," in *ICCCN '08*, St. Thomas, VI, Aug. 2008.
- [12] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation service for ad-hoc sensor networks," *SIGOPS Operating System Review*, vol. 36, no. SI, pp. 131–146, Dec. 2002.
- [13] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," in *ICC '04*, Paris, France, June 2004, pp. 3640–3645.
- [14] K.-W. Fan, S. Liu, and P. Sinha, "Scalable data aggregation for dynamic events in sensor networks," in *SenSys '06*, Boulder, CO, Nov. 2006, pp. 181–194.
- [15] W. Zhang and G. Cao, "DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1689–1701, Sept. 2004.
- [16] J. Feng, D. Eager, and D. Makaroff, "Asynchronous data aggregation for real-time monitoring in sensor networks," in *IFIP Networking '07*, Atlanta, GA, May 2007, pp. 73–84.
- [17] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *SenSys '04*, Baltimore, MD, Nov. 2004, pp. 250–262.
- [18] S. Gobriel, S. Khattab, D. Mosse, J. Brustoloni, and R. Melhem, "RideSharing: Fault Tolerant Aggregation in Sensor Networks Using Corrective Actions," in *SECON '06*, Reston, VA, Sept. 2006, pp. 595–604.
- [19] S. Motegi, K. Yoshihara, and H. Horiuchi, "DAG based in-network aggregation for sensor network monitoring," in *SAINT '06*, Phoenix, AZ, Jan. 2006, pp. 292–299.
- [20] J. Feng, D. Eager, and D. Makaroff, "Aggregation protocols for high rate, low delay data collection in sensor networks," in *IFIP Networking '09*, Aachen, Germany, May 2009, pp. 26–39.
- [21] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 107–123, Jul. 2003.