Intelligent Sensor Activity Scheduling for Event Monitoring Networks

Sumathi Seetharaman, Ali A. Minai
University of Cincinnati

Abstract—**Large-scale sensor networks are being envisioned and applied in a wide range of event monitoring tasks like forest fire detection, enemy movement tracking in a battle field, crack detection on a bridge etc. Field coverage is a critical issue for such applications since the sensor network must identify the event quickly and accurately. In this paper, we compare several decentralized, self-organized methods for scheduling nodes to obtain effective field coverage. The focus is on assessing how well individual nodes in the network can locally estimate and optimize their schedules in order to achieve complete global coverage and a longer network lifetime at a lower energy cost.**

***Index Terms*—Wireless Sensor Networks, Distributed Algorithms, Self-organization**

I. INTRODUCTION

Advances in MEMS (Micro-Electro-Mechanical Systems) technology have enabled the development of extremely small multi-functional sensing devices. Due to their miniature size, hundreds or thousands of these micro-sensors can be scattered over an environment to sense intricate information. The sensors can communicate locally with each other and self-organize to form a functional sensing network. Some key advantages of such pervasive sensor networks are: a) Ubiquitous, non-intrusive nature; b) Random deployment with limited or no pre-established infrastructure; c) Minimal supervision since nodes can self-organize into a functional network through local collaboration, d) Cost effectiveness; e) Flexibility; f) Scalability; g) Simple expandability; and h) Robustness. Application areas proposed so far include habitat monitoring, space exploration, precision agriculture, military surveillance, factory instrumentation, weather monitoring, smart spaces, medical, structural health monitoring, seismic and acoustic data gathering applications etc.

Individual micro-sensor nodes have severe energy constraints, limited memory, rudimentary computing resources, and are prone to failure. They can perform a limited set of functions: a) sense the environment, b) transmit and receive data packets, c) perform simple information processing and d) stay inactive to conserve battery.

Primary function of event monitoring networks is to maintain *coverage*, i.e., to ensure that every location in the monitored environment is being sensed at all times. Another objective is to conserve energy, since nodes typically have limited energy resources. Nodes in a large-scale sensor network are usually deployed randomly with relatively high density. Thus, even though each node can sense only in a limited region, the sensing regions of neighboring nodes overlap significantly. The *coverage problem* is to coordinate the *-sense-sleep-* activity cycle of nodes in a way that ensures sensing coverage while minimizing energy consumption.

In this paper, we describe two intelligent scheduling schemes and compare their performance with some successful scheduling and density control algorithms. Based on the results, we draw some interesting conclusions regarding the ideal design space for each approach.

II. COVERAGE ALGORITHMS

Ideally, a coverage algorithm must be simple and efficient, consume the minimum possible energy for initialization and functioning overhead, be adaptive to dynamic environmental and topological changes (node failures), self-configure using local information, and scale well with network size.

The coverage problem can be modeled as the classic set covering problem using circular discs. In randomly deployed large-scale sensor networks, one has to address network lifetime issues in addition to field coverage. Thus, the coverage problem can be taken a step further, asking for the maximum number of mutually exclusive subsets that can cover the area in turns [2].

A number of interesting solutions to the coverage problem have been explored and can be broadly categorized as centralized and distributed schemes. Centralized scheduling approaches like integer linear programming [5], blue noise spatial sampling [3], etc. cannot be used in large scale sensor networks due to excessive communication overhead in relaying topology information. Due to distributed sensing behavior and the dynamic topology of the network, decentralized algorithms are better suited to adapt the node schedules to the local environment.

Distributed approaches to solving the coverage problem can be classified as: a) *Scheduling schemes* - graph coloring [4],

differential coverage algorithm [7] etc and b) *Density control schemes* – PEAS [8], OGDC [9], RACP [1], sponsored sector scheme [6] etc. Scheduling schemes partition the set of nodes into subsets called *covers* such that each subset can completely cover the field. These covers share the burden of monitoring by taking turns at being active. In contrast, density control schemes seek to maintain a uniform density of active nodes by activating an initial cover and activating new nodes as needed to replace those that run out of energy.

OGDC [9] is designed for very dense networks and nodes are added into the cover by a sequential location-based activation scheme. While this scheme may produce a good initial layout of active sensors, it is unclear how the scheme will adapt to failures. Since the density of the network reduces with time due to node failures, the probability of finding a node at a desired location diminishes with time and the scheme may yield suboptimal solutions. Graph coloring schemes [4] take time to converge to a good solution. The differential coverage algorithm proposed by Ting [7] exhibits good performance with a low overhead and is shown to perform better than the overlap-based sponsored sector scheme [6].

There has not been any formal comparison of scheduling and density control techniques to our knowledge. Density control schemes can be implemented using simple timer based mechanisms whereas distributed scheduling schemes employ a more elaborate initialization phase and failure adjustment phase requiring more energy. We compare our work with an efficient distributed scheduling algorithm presented in [7] and a robust density control scheme presented in [8]. These two schemes are described here briefly for the sake of convenience.

In the differential coverage algorithm, sensing periods are time sliced into intervals of duration T. Each node selects a reference time point (t_{ref}) between 0 and T randomly and broadcasts it. After receiving the reference times of neighboring nodes, each node computes the smallest schedule for each sensing field within its sensing region. A sensing field is an area within a node's sensing region that is covered by a unique set of sensors. For example, if field f_a in the sensing region of s_i is covered by one other node s_j with reference time $t_{(j,ref)}$, then the field schedule for node s_i is evaluated as:

$$t_{(i,front)} = \{T + t_{(i,ref)} - t_{(j,ref)}\}/2$$
$$t_{(i,end)} = \{t_{(j,ref)} - t_{(i,ref)}\}/2$$
$$Schedule_{(i,f_a)} = [t_{(i,ref)} - t_{(i,front)}] - [t_{(i,ref)} + t_{(i,end)}]$$

assuming $t_{(j,ref)} > t_{(i,ref)}$. Finally, the sensing schedule determined by overlapping all field schedules is broadcasted. To optimize further, each node tries to shrink its schedule as far as possible, ordered by the length of node schedules to avoid conflicts. Although this algorithm is simple and efficient, the random selection of reference times forces sensors to remain active for longer periods than necessary. We propose an alternative random selection algorithm detailed in the next section which produces better results with the same communication overhead.

In density control schemes, PEAS [8] is an extremely adaptive and robust protocol that works by nodes adjusting their periods of "sleeping" (inactivity). Sleeping nodes periodically wake up and probe their neighborhood to find out if any nearby sensor is active. If the querying node is not necessary to provide coverage, it dynamically adjusts its sleeping period based on the observed probing rate of nearby active nodes and enters the sleep mode. Otherwise, it joins the

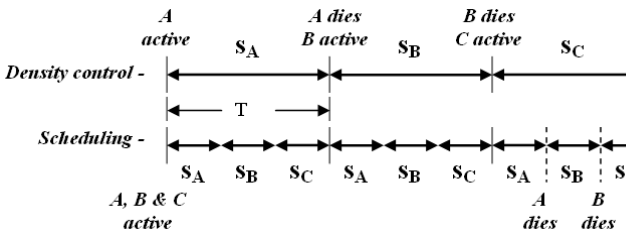

Fig. 1. Ideal scheduling scenario

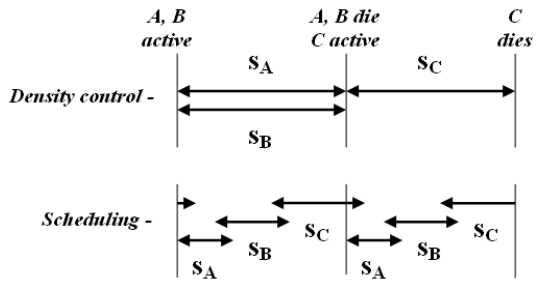

Fig. 2. Practical scheduling scenario

covering set of nodes. The probing range controls the coverage redundancy while the rate of probing determines the average latency in replacement of a failed node.

Fig. 1 shows that, ideally, a region covered by nodes A, B and C should be covered for the same duration by both density control and scheduling schemes. However, due to density variations and nonuniform overlap, suboptimal solutions may be generated by both schemes as shown in Fig. 2. In density control schemes, two nearby nodes may be forced to stay active at the same time and in scheduling schemes, they may be forced to stay on for a longer period than required. Both these situations result in sensing redundancy which in turn reduces network lifetime.

III. INTELLIGENT SCHEDULING

None of the coverage schemes discussed above explicitly consider the resulting spatial distribution of coverage. The

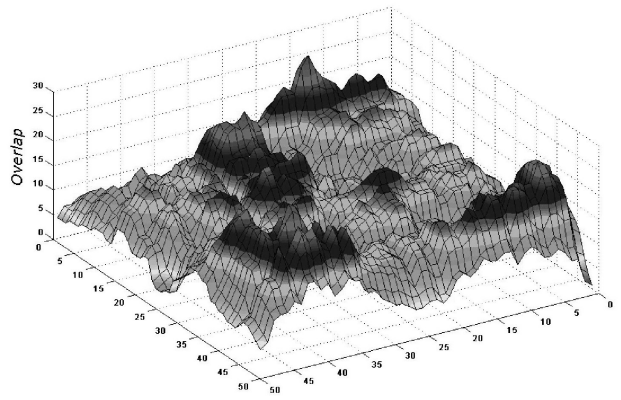

Fig. 3. Overlap distribution in the field with 500 sensors randomly distributed in a 50m x 50m field

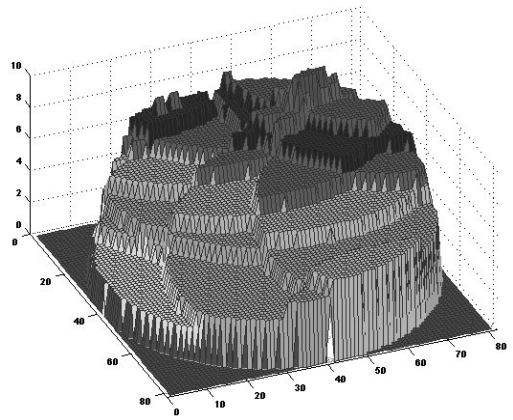

Fig. 4. Coverage distribution within a node's sensing region

random deployment of nodes causes nonuniform placement of nodes in the field and the number of nodes covering individual points in the field varies significantly. The surface of the coverage landscape in Fig. 3 indicates wide variations in the extent of coverage from region to region. Interestingly, the coverage landscape within the sensing region of a node (Fig. 4) exhibits similar surface features. It can be observed that overlap between the sensing regions of neighboring nodes divides the area into multiple *sensing fields* covered by different numbers of sensors.

We address the coverage problem by allowing the nodes to make simple decisions based on observable local topology. Each node acts greedily and attempts to minimize its schedule as far as possible. However, the extent to which a node can shrink can its active sensing time depends on its *weakest field*, defined as a field within its sensing region that is covered by the smallest number of sensors. Let f_b be the weakest field for node s_j covered by one other node s_k. To cover this sensing field efficiently, nodes s_j and s_k each need to be active for duration $T/2$ in each sensing round of length T. The *weakness degree* of each node is the number of nodes covering its weakest field. Formally, let $f'_{(i,\ \text{sense})} = \{ f_{(i,1)},\ldots f_{(i,g)} \}$ be the set of sensing fields, and $N_{(i,cover)} = \{ n_{(i,1)},\ldots n_{(i,g)} \}$, where $n_{(i,j)}$ is the number of nodes covering field $f_{(i,j)}$. Define:

$$n_{(i,weak)} = \min(N_{(i,\text{cover})})$$

$$a_{(i,weak)} = \sum_{j|n_{(i,j)}=n_{(i,weak)}} a(f_{(i,j)})$$

where $a(f_{(i,j)})$ is the area of region $f_{(i,j)}$. Thus, $a_{(i,weak)}$ is the total area that is most weakly covered, and $n_{(i,weak)}$, termed the *weakness degree* of i is the number of sensors covering it.

We can make simple deductions based on a node's weakness degree:

1) Nodes with weakness degree $n_{(i,weak)} = 1$ need to be active at all times to provide complete coverage to their weakest field.
2) Ideally, a node with weakness degree $n_{(i,weak)} = 2$ needs to be active for half the sensing round.
3) Nodes may have more than one weakest region in their sensing field. Each such region would be covered by a different though possibly overlapping group of sensors.
4) A node may be forced to work for longer periods than necessary due to nonuniform overlap in node sensing areas. Fig. 5 shows three nodes A, B and C sharing weak regions with each other. Due to inter-dependence, it is impossible to work out a mutually exclusive schedule and one of the nodes is forced to extend its schedule.
5) Weakness is relative. The weakest region of a particular node may be a relatively strongly covered region for another node.

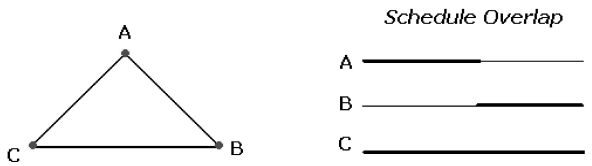

Fig. 5. Unavoidable schedule extension due to overlap

6) Given that a node has weakness degree d (where $d > 1$) there must be at least 1 neighboring nodes that have weakness degree of d or lower.

Our intelligent scheduling approach is based on the idea that the sensors covering the most weakly covered regions have the greatest need to conserve energy, since they are the least dispensable nodes. Also, since nodes lying in sparse regions need to be active longer, the neighboring nodes can exploit this information and conserve their energy.

To provide a baseline, we consider a Random Scheduling Algorithm (RSA), where all nodes have the same duty cycle.

Each node randomly selects m slots out of a total of k slots available in each sensing round. We describe the two intelligent scheduling algorithms below.

A. Adaptive Scheduling Algorithm (ASA)

This algorithm enhances the RSA by incorporating a local decision making step that allows nodes to adaptively change their duty cycle to provide sufficient coverage for their sensing region. An initial *neighbor discovery phase* is introduced where nodes announce their positions and record the locations of nearby nodes. Using this information, each node can construct the coverage landscape within its sensing region and evaluate its weakness degree. Simply put, nodes with lower degrees must stay on for longer durations because they have fewer nodes to share the burden in their weakest sensing regions.

Given its weakness degree, each node locally estimates its schedule to ensure complete coverage of its sensing field with probability p_{cov}. Since each node independently selects the time slots for which it is active, the estimated duty cycle is higher than necessary to ensure 100% coverage.

Mathematically, each node s_j decides to be active in a particular slot t_k with probability d_j. If n_j be the weakness degree of node s_j, then the probability that at least one node covering s_j's weakest region is active during slot t_k is given by:

$$p(\text{cov}er \geq 1, t_k) = 1 - (d'_j)^{n_j} \quad (1)$$

where $d'_j = 1-d_j$. If the field is completely covered (for all slots) with probability p_{cov}, then the node s_j can estimate its duty cycle d_j using the below mentioned equation where t is the total number of slots in a sensing round.

$$p_{\text{cov}} = \{1 - (d'_j)^{n_j}\}^t \quad (2)$$

Probability p_{cov} may be pre-programmed or supplied by the end user in real time. The value of p_{cov} determines the quality of coverage provided by the scheme. A low value results in significant lapses in coverage and a high value causes redundant sensor operation. Using test simulations, p_{cov} can be tuned to produce desired level of coverage with low redundancy.

This topology sensitive schedule selection allows nodes lying in denser regions to shrink their on-times and conserve energy while ensuring the sparse regions are adequately covered. The random selection of active slots still causes some sensing overlap and an optimization procedure to enhance efficiency is discussed later in the paper.

B. Collaborative Scheduling Algorithm (CSA)

Due to nonuniform coverage distribution in the field, the weakness degrees of neighboring nodes can vary significantly. The CSA attempts to take this into account when nodes decide their schedules. For example in Fig. 6, node s_A has weakness degree 2, but the other node, s_B, covering its weakest region has a weakness degree of 1. This means that s_B will have to stay on throughout the sensing round to cover its weakest region, and the weakest region of s_A will be covered completely by s_B. This leaves s_A free to decide its schedule based on its next weakest region, resulting in a lighter schedule.

As discussed earlier, a node may have many weak fields. Therefore, each node needs to check all of its weak fields before deciding to update its weakness degree. Decision based

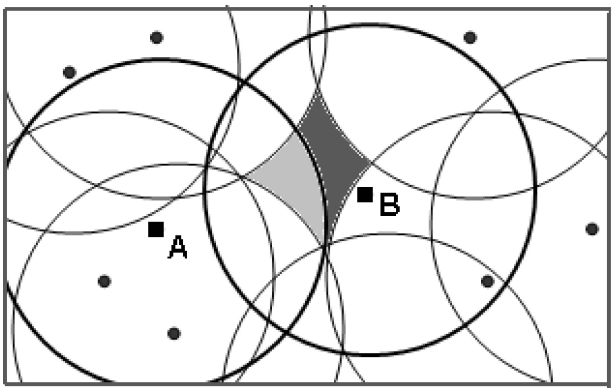

Fig. 6. Collaborative Algorithm: Example

on one weak region alone may result in reduced coverage for another. After determining that all the weakest regions are being covered adequately by neighboring nodes, the next weakest region can be identified and broadcasted.

Allowing nodes to asynchronously adjust their schedule without proper coordination results in conflicts and lower coverage. Therefore following the idea that weaker nodes need to conserve as much energy as possible, each node compares its weakness degree with that of its neighbors. Nodes with the lowest weakness degree in a local neighborhood are the most *critical* since they need to stay active the longest in order to cover their weakest regions. Therefore, each node prioritizes itself according to its weakness degree $n_{(i,weak)}$ and allows weaker nodes to minimize their schedules first. Thus, the nodes locally determine the negotiation order using the neighborhood information gathered in the previous steps. In case of a conflict, the nodes with the largest weak area $a_{(i,weak)}$ have a higher priority and therefore attempt first. Once a node broadcasts its final weakness degree, the neighboring nodes update their neighborhood lists and adapt their decisions accordingly.

Both ASA and CSA have a low communication overhead. For a network with n nodes, the adaptive and collaborative scheme require a total of n and $2n$ broadcasts respectively.

C. Greedy Optimization

While ASA uses absolute weakness to adapt on-times, CSA aims at exploiting the relative weakness of nodes to shrink active sensing time further. In this next optimization step, we attempt to minimize the schedule overlap between neighboring nodes due to randomized slot selection. This dual approach ensures that the weakest node is functioning optimally while, at the same time, the neighboring nodes exploit its longer on time to conserve their energy.

As a preprocessing step, each node checks all its sensing fields for complete coverage. If there are any uncovered slots, the node extends its own schedule to provide 100% coverage within its sensing region. This additional step is added before the greedy optimization procedure to ensure maximum possible coverage of the field.

Schedule optimization is carried out using a sequential algorithm in which nodes with more critical values of $\{n_{(i,weak)}, a_{(i,weak)}\}$ attempt to shrink their schedules first. This procedure is similar to CSA: Nodes use the neighborhood list containing

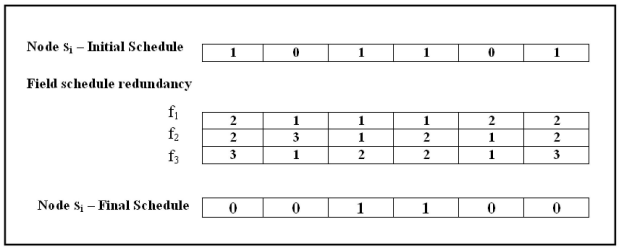

Fig. 7. Greedy schedule optimization procedure. Node s_i attempts to shrink its on-time by exploiting neighbor schedules.

the weakness parameters of nearby nodes to determine the negotiation sequence and their position in this sequence. When it is the turn for a node to optimize, it checks all of its sensing fields for slots that are redundantly covered and shrinks its on-time accordingly.

In Fig. 7, we illustrate this greedy optimization using a simple example. Consider a node s_i having three sensing fields (f_1, f_2 and f_3) in its sensing region. The sensing rounds are assumed to have 6 slots. Schedule redundancy is defined as the total number of nodes covering each slot in the schedule. Field schedule redundancy is compared with the node s_i's active schedule to identify any slots that are being redundantly covered. Node s_i turns off a slot t_k only when all the fields within its sensing region are redundantly covered by other nodes in that slot. In the diagram shown below, slot t_1 and t_6 are both covered by other nodes for all three fields. Therefore, the final active schedule for node s_i is minimized.

Since each node determines its rank or priority in its local neighborhood using $n_{(i,weak)}$, the latency in determining the optimal schedule depends on the number of neighboring nodes. For a given communication range, latency increases with node density.

TABLE I
NETWORK PARAMETERS

Parameter	Value
Field size	50m X 50m
Number of nodes	500
Sensing radius	4m
Communication radius	8m
Simulation time	50,000s

IV. SIMULATIONS AND DISCUSSION

For simulation, we assume that the nodes are aware of their location coordinates. This can be accomplished easily using GPS enabled nodes or a distributed localization algorithm with considerable accuracy. Each sensing round consists of 20 equally sized slots and the nodes are assumed to be time-synchronized to acceptable precision. The energy model used is derived from hardware specifications in [10]. Table 1 specifies the parameter values used in the simulation. The performance of the algorithms is assessed in two scenarios. In the ideal scenario, nodes are assumed to have infinite energy and are immune to failure. This case represents the best case scenario for the schemes and provides a comparison regarding their intrinsic merit. In the non-ideal scenario, nodes have finite energy and fail to function once energy is depleted. This test case evaluates the robustness of the scheduling schemes, assessing performance over time as well as overall network lifetime.

Fig. 9 provides a detailed breakup of coverage provided by RSA for regions with specific weakness degrees. As expected, the sparser regions are adequately covered (> 80%) only at very high duty cycles. In contrast, Adaptive and Collaborative algorithms provide near 100% coverage for p_{cov} set to 0.6 (Fig. 4). However, CSA's coverage is slightly lower due to the fact that as nodes revise their weakness degrees, their on times decrease and random scheduling may sometimes result in low coverage.

Fig. 10 displays the distribution of the node's sensing times for ASA, CSA, and DSA before and after optimization over 15 simulations. The reduction in schedules due to collaboration can be observed as the distribution shifts towards lower active times. It can be seen that after optimization both adaptive and collaborative algorithms (oASA and oCSA) minimize node schedules to a greater extent compared to the optimized differential algorithm(oDSA).

Field Redundancy is a measure of the extent of sensing overlap in the network. We define redundancy as the average number of additional nodes covering each slot. While some level of redundancy is unavoidable, it serves as a good measure of algorithm inefficiency. Instead of simply measuring overall field redundancy, we take a qualitative approach. We

classify sub regions according to their weakness degree and evaluate the redundancy for each such region. Although, only results with 500 nodes are presented here, similar behavior is observed at all densities (250-1000). It can be seen from Fig. 11, that as the coverage degree of the field increases, the average redundancy produced by all the schemes is higher. Collaboration lowers the redundancy produced by ASA. DSA displays much higher redundancy compared to ASA and CSA. This is due to the fact that in DSA, nodes are forced to have continuous schedules which lead to greater overlap. Also, the optimization procedure drastically reduces the redundancy displayed by all three schemes.

After optimization, ASA and CSA perform almost similarly. The reason for this is that the underlying approach in CSA is mirrored in the greedy optimization algorithm as well. In both the schemes, weaker nodes attempt to reduce their on-time. Hence, this optimization on ASA produces equally good results as CSA. Therefore, the additional communication expense of n transmissions in CSA can be avoided by simply employing oASA. Even after optimization, oASA and oCSA perform better than oDSA.

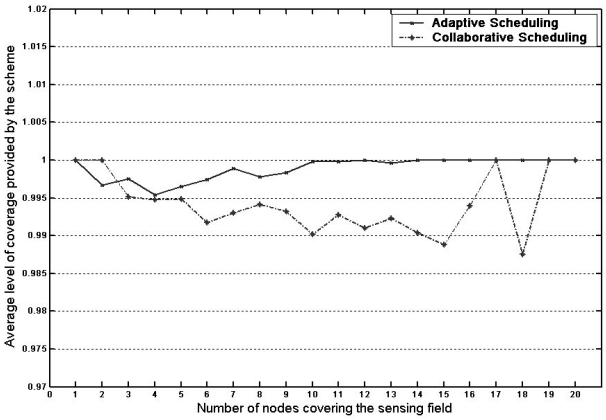

Fig. 8. Adaptive Scheduling vs. Collaborative Scheduling:- Comparison of the average level of coverage provided by the schemes to regions with the same weakness degree.

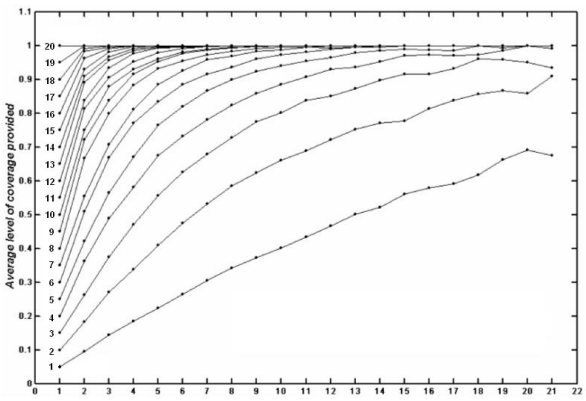

Fig. 9. The average level of coverage provided by RSA to regions with a given weakness degree are plotted. RSA is evaluated for all duty cycles. The numbers next to the curve represent the number of active slots out of 20.

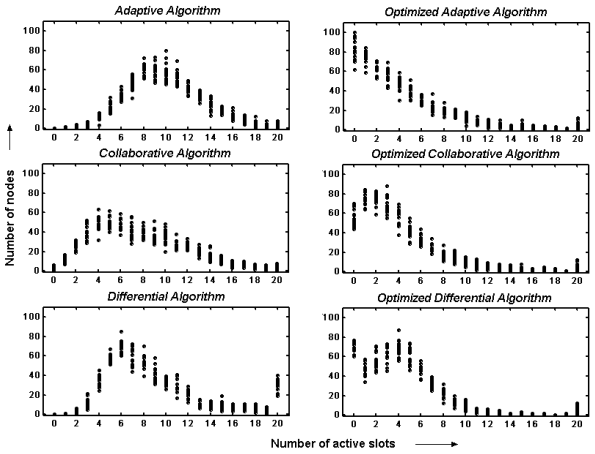

Fig. 10. Detailed comparison of node on-time distributions of ASA, CSA and DSA, before and after optimization over 15 runs

To assess the level of coverage provided by the schemes in the whole field, the percentage field area with a given degree of coverage is evaluated and plotted in Fig. 12. The degree of coverage in a sensing region at a given instant is defined as the average number of nodes covering that region during a sensing round. It can be observed that after optimization (Fig. 13), nearly 60% of the field area is covered by only one node at a given time.

A. *Negotiation Strategy*

It is easy to see, intuitively, that the duty cycle of nodes decreases as density increases. It has been proven in previous research that this rate of reduction gradually decreases with increase in density and finally saturates [1]. The reason for such a behavior can be explained easily using our critical graph model described in Section III. With increase in density, two conflicting factors come into play. Increase in density immediately implies that more nodes are available to cover a particular region. Therefore, individual contribution of nodes should scale down as well. However, with increase in density, the number of critical links a node forms also increases due to excessive overlap. This heavy interdependence among nodes forces them to stay on longer than necessary. As node density increases, the latter effect balances the former and as a result the schedule reduction ceases after a point.

This behavior points us to another interesting aspect of our algorithm: In the collaborative scheduling algorithm, the schedule shrinking strategy must exhibit similar behavior. Fig. 14 illustrates this effect clearly. The improvement from ASA to CSA is measured in terms of the normalized increase in the number of nodes with lower on-times. As noted from the figure, at lower densities the improvement increases steadily. At the same time, the rate of increase gradually decreases and further

improvement ceases at higher densities.

B. Scheduling versus Density control

We also investigate the relative advantage / disadvantages of scheduling and density control. For this purpose, we compare the optimized ASA with the PEAS algorithm [8] using 1000 nodes since PEAS is designed for higher densities. The probing range in PEAS is set to ensure near 100% coverage [8]. It can be observed that PEAS exhibits gradual degradation in performance although it activates a larger number of nodes at a time and provides a lower level of coverage than optimized ASA (Fig. 15). On the other hand, optimized ASA provides nearly 100% coverage for a longer period and then undergoes abrupt degradation. In Fig. 15, *PEAS: Ideal* refers to the performance achievable with the network topology at any given instant and PEAS refers to the actual performance exhibited by the scheme.

As density decreases due to node failures, the coverage performance of PEAS deteriorates even further. This is because PEAS ensures that the distance between any two working nodes is at least the probing distance, but does not guarantee complete coverage. In sparse regions blind holes may appear and the algorithm is unaware of the existence of such spots. PEAS is more effective in prolonging network functioning and shows graceful degradation with node failure whereas the adaptive scheme attempts to maximize coverage for as long as possible and is a best effort scheme.

For high density networks with fragile nodes, scheduling schemes are inappropriate since the communication overhead involved in maintaining and regularly updating neighborhood topology will increase considerably. However even at high densities PEAS activates more nodes at a time to ensure near 100% coverage.

The schedule shrinking behavior noted earlier in this section indicates that at higher densities, scheduling schemes alone might not be very efficient in eliminating redundancy. It might be advantageous to have a density control scheme working in parallel to maintain a desired density of active nodes in the network. Therefore, an optimal solution to the coverage problem might lie in a combination of scheduling and density control approaches.

V. CONCLUSIONS AND FUTURE WORK

We proposed and evaluated the performance of two randomized algorithms: Adaptive scheduling and Collaborative scheduling. The adaptive scheme highlights the benefits of intelligent local decision-making by the nodes and the collaborative scheme causes the nodes to negotiate a smarter schedule by mutual negotiation. The algorithms were shown to perform better than the differential coverage algorithm. The schedule improvement from ASA to CSA was analyzed and quantified over a range of densities to highlight the effect of sensing region overlap. We compared our scheme to a density control algorithm PEAS and observed that density control schemes fare better in extending network lifetime but at the

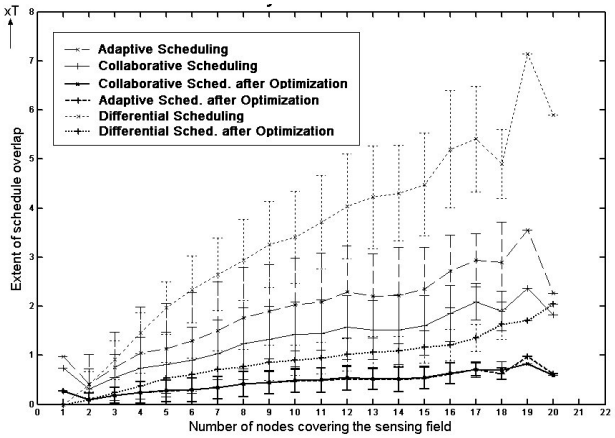

Fig. 11. Field redundancy distribution with 500 nodes in the network

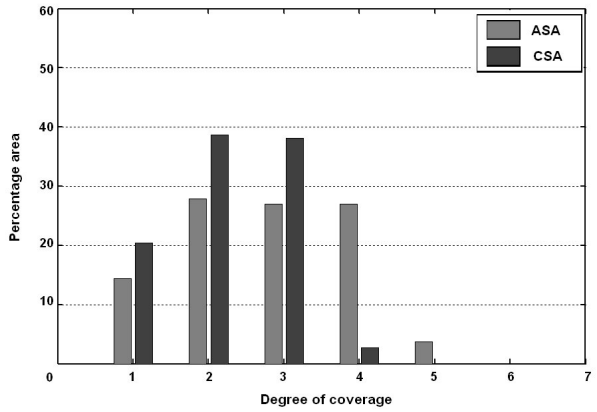

Fig. 12. Coverage degree distribution in the field with ASA and CSA

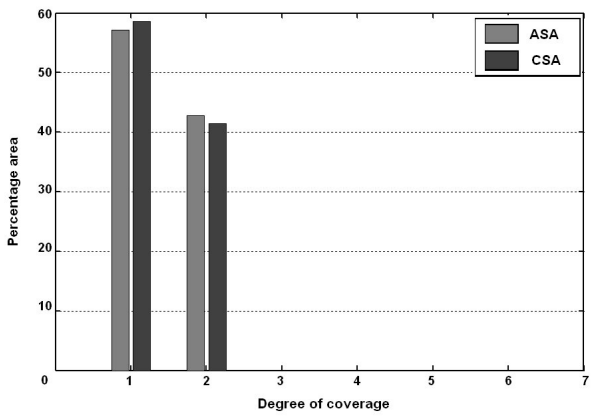

Fig. 13. Coverage degree distribution in the field with oASA and oCSA

cost of coverage quality, whereas the adaptive scheme achieves high levels of coverage for a longer duration but undergoes catastrophic degradation.

ACKNOWLEDGMENT

The authors would like to acknowledge Prasanna Ranganathan for his contribution to this paper.

REFERENCES

[1] Hsin, Chih-fan, and Mingyan Liu, "Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms", Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN), (April 2004).

[2] Huang, Chi-fu, and Yu-Chee Tseng, "The coverage problem in a wireless sensor network", Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, ACM Press (2003), 115-121.

[3] Ignjatovic, Zeljko, Mark Perillo, Wendi Heinzelman, and Mark. Bocko, "An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique", 2nd IEEE Upstate New York Workshop on Sensor Networks, (October 2002).

[4] Ko, Bong-Jun, Keith Ross, and Dan Rubenstein, "Conserving energy in dense sensor networks via distributed scheduling", Extended abstract, Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03), (March 2003).

[5] Meguerdichian, Seapahn, and Miodrag Potkonjak, "Low power 0/1 coverage and scheduling techniques in sensor networks", Tech. Rep. no. 030001, UCLA, (January 2003).

[6] Tian, Di, and Nicolas D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks", Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, (2002), 32-41.

[7] Yan, Ting, Tian He, and John A. Stankovic, "Differentiated surveillance for sensor networks", Proceedings of Sensys, (August 2003).

[8] Ye, Fan, Gary Zhong, Jesse Cheng, Songwu Lu, and Lixia Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks", Proceedings. 23rd International Conference on Distributed Computing Systems (IEEE ICDCS), (2003), 28-37.

[9] Zhang, H., and Jennifer C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks", NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, (February 2004), Invited paper.

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors", SIGPLAN Not. 35 (2000), no. 11, 93-104.

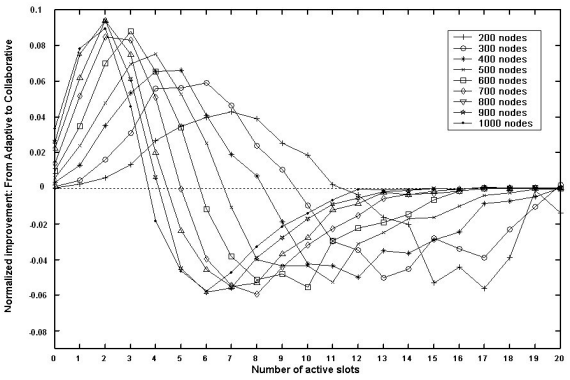

Fig. 14. Schedule improvement from ASA to CSA. It is measured as the normalized decrease in node on-times due to node collaboration.

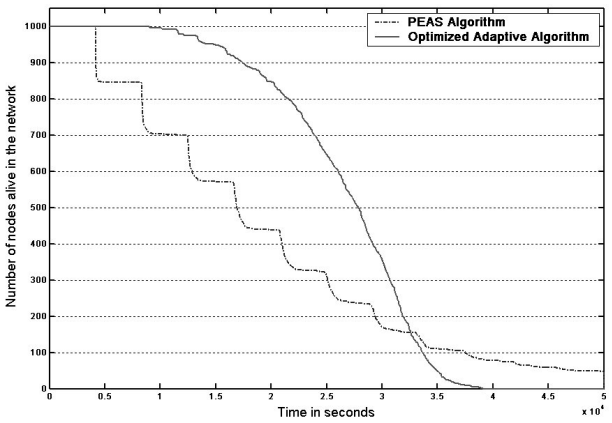

Fig. 16. Comparison of the total number of nodes alive in the network at a given instant of time