DCA – A Distributed Channel Allocation Scheme for Wireless Sensor Networks*

Kaushik R. Chowdhury, Pritam Chanda, Dharma P. Agrawal, and Qing-An Zeng
University of Cincinnati, Cincinnati, OH 45221-0030
Email:kaushir,chandap,dpa,qzeng@ececs.uc.edu

Abstract - **This paper introduces a distributed channel assignment scheme (DCA) for wireless sensor networks. Recent developments in sensor technology, as seen in Berkeley's Mica2 Mote, Rockwell's WINS nodes and the IEEE 802.15.4 Zigbee, have enabled support for single-transceiver, multi-channel communication. DCA exploits this capability by assigning optimally minimum channels in a distributed manner in order to make subsequent communication free from both primary and secondary interference. The main contribution of this paper is solving the channel assignment problem, also named as the 2-hop coloring problem, in which repetition of colors occurs only if the nodes are separated by more than 2 hops. Our work achieves legal coloring under energy constrained conditions for sensor networks and significant performance improvements are shown when compared with similar existing schemes. Finally, we validate our approach through extensive analysis and simulation results.**

Keywords: **Channel allocation, Distributed algorithm, Graph coloring, Interference, Protocols, Sensor networks.**

I. INTRODUCTION

With improvement in sensor hardware, support for multi-channel communication is already in an advanced stage of implementation. Berkeley's third generation Mica2 Mote has an 868/916 MHz multi-channel transceiver [1]. In Rockwell's WINS nodes, the radio operates on one of 40 channels in the ISM frequency band, selectable by the controller [2]. The new IEEE 802.15.4 standard, popularly called as Zigbee, defines 11 channels at 868/915 MHz and 16 channels at 2.4 GHz [3].

In our work, we propose a distributed channel allocation scheme to exploit this multi-channel capacity in sensor networks while taking into interference avoidance. If allocated channels do not repeat within 2-hops of a node, both primary (sender and receiver using same channel) and secondary (communication between sender and receiver interfering with another pair-wise data transfer) interference [4] can be avoided. In addition, our channel assignment algorithm has to ensure minimum energy consumption for its operation and use the fewer number of channels separating them as much as possible in the frequency band, thus achieving high frequency reuse. The problem of channel allocation is similar to the code assignment problem in Code Division Multiple Access (CDMA) networks that eliminates collisions through spread spectrum techniques and orthogonal codes. Earlier work on CDMA code allocation have attempted to solve this problem as a graph coloring problem where colors can be repeated only at three hops or more, unlike traditional graph coloring as surveyed in [9] (and the references therein). We shall henceforth refer to coloring under this constraint as the NP complete 2-hop coloring problem. Centralized code assignment schemes are unsuitable when directly applied to sensor networks as it may be infeasible to establish direct communication between the base station (BS) and the individual sensor nodes. Past work has also involved presenting optimal 2-hop coloring algorithms for special topologies like the ring, bus, chain, tree and hexagonal grids. In the analysis of the Distributed Coloring Algorithm (DCA), we consider random deployment of the sensor nodes. Recent work, including the algorithm proposed in [10] reduces, under assumptions of ordering, to the Hidden Primary Collision Avoidance (HP-CA) suggested in [7] and modified in [5, 8]. In [7], when a node chooses a color, it is propagated to its two hop neighbors having an ID lower than itself. The simple modification of making a node wait till it hears from all nodes having a higher ID and within its two-hop range before announcing its own non-conflicting color provides an elegant solution. We call this HP-CAM with '*M*' for modified. In [6], a node chooses a unique ID and broadcasts it amongst its two hop neighbors. In case of a conflict, the information is conveyed back to the transmitting node and new ID is chosen. Though simple, this process involves large message passing as $n-1$ re-tries may be required for each node in the worst case. In HP-CAM, each node is aware of its 2-hop information and generates an ordering based on node weight. This results in reduced number of messages as there is no color conflict once an assignment has been done. We thus compare the DCA with HP-CAM while evaluating the performance in this paper.

The rest of this paper is organized as follows. Our DCA algorithm is presented in Section II. In Sections III and IV, we provide detailed analytical and simulation results that measure the performance of the DCA respectively. Concluding remarks and directions for future research can be found in Section V.

II. DCA ALGORITHM

In this section we describe the DCA, that colors the network graph within the constraints of 2-hop coloring. We list our assumptions and provide a discussion on each of them:

1. A 1-hop clustering structure is in place: The non-linear relationship between energy and distance makes a single bit transmission more energy efficient using several short, intermediate hops instead of one longer hop [12]. Clustering allows sensors to communicate data over smaller distances.

* This work has been partially supported by the Ohio Board of Reagents' Doctoral Enhancement Funds.

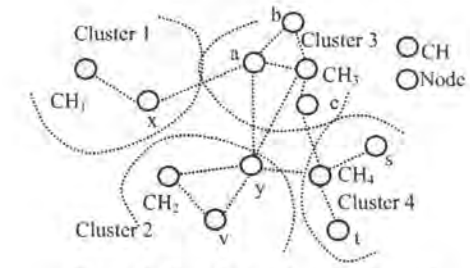

Figure 1. Sample topology. Clusters 1 and 2 are hidden from each other.

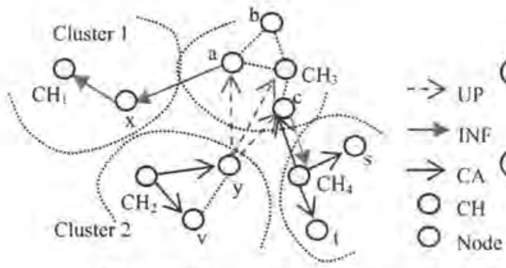

Figure 2. Types of messages and their flow.

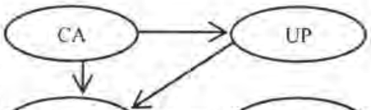

Figure 3. Message dependencies: The receipt of one type of message generates another.

We have adapted the generalized weight based clustering scheme presented in [11] with the weight being decided by connectivity and ties resolved with node ID. Each clusterhead (CH) knows the weights of the CHs of all nodes that are within two hop range of the nodes in its own cluster. The influence of the choice of the clustering algorithm on the coloring process is explored in Section IV.

2. The sensor network is assumed to be static during the coloring process. No new nodes are added and nodes do not die out while coloring is in progress.

3. A CSMA/CA mechanism is in place during algorithm operation. At the point of termination of the algorithm, the channel is partitioned into disjoint sets and is essentially collision free, assuming no adjacent channel interference.

Explanation of the algorithm:

We define the term '*neighbor*' set of a cluster i, NS_i, as the set of all those nodes, n, such that n is at most two hops away from any node in i and $\notin i$, e.g. in Figure.1, considering cluster2, $NS_2 = \{x, a, b, s, t, c, CH_3, CH_4\}$. The problem of color allocation is essentially coloring of a node such that there is no conflict with the already assigned colors of the nodes in the *neighbor set*. In particular, consider two nodes, x and y linked through a and each belonging to a different cluster. This configuration poses a difficult task in assignment of non-conflicting color and we define it as the *hidden cluster problem*. This necessitates the longest chain of color information propagation across cluster boundaries.

The DCA begins at a point where each CH is aware of its own weight, $w(CH)$, as well that of the other CHs that serve its *neighbor* set, i.e., CH_2 knows $w(CH_1)$, $w(CH_3)$ and $w(CH_4)$. A CH proceeds to color its own cluster only when color assignment by the larger weight CHs of the *neighbor* nodes is complete. If $w(CH_2) > w(CH_3) > w(CH_4) > w(CH_1)$, CH2 starts coloring first followed by CH3. On their completion, the color assignment is propagated and thus CH1, CH4 begin coloring simultaneously ($x, CH_1 \notin NS_4$). It should be noted that there is no possible color conflict in clusters 1 and 4, as the *neighbor* set of one has no node from the other cluster.

Each node in cluster i maintains a color table (CT) containing a list of its *neighbor* nodes belonging to a CH having an ID higher than its own CH, e.g., $CT_y = \Phi$ while $CT_a = \{v, y, CH_2\}$. As a node receives color information about these nodes, this list is updated and all these colors are forbidden for the node itself. In addition, the CH has a table, which we call as the cluster color table (CCT). The CCT has an entry for each node in its cluster and the colors forbidden for it are continuously updated, as in the case of CT. A CH starts coloring only when this CCT is full, i.e., all the forbidden colors for each node in its cluster are available. Each CH performs this check at the receipt of a message, in order to check whether the information contained in the CCT is sufficient to start the coloring process.

The DCA is entirely message driven, i.e. an action taken by a node is a function of the type and information contained in the incoming packet. We define three types of messages: The *channel assignment* message that is sent by the CH letting a node know of its assigned color, the *update* message aimed at 1-hop *neighbors* to make them aware of a forbidden color, and the *information* message which alleviates the hidden cluster problem while notifying the CH of a forbidden color for a node in its cluster. We now describe, through an example (Figure 2), the actions taken by nodes for different control messages. We exclude a formal pseudocode representation for want of space.

Channel Assignment (CA) message: The *CA* message consists of an ordered pair of the type (*node ID*, *assigned color*) for each cluster member. A node first identifies whether the message was sent by its CH. It then extracts its own color and that of nodes 1 hop away from it and member of the same cluster. Thus node y identifies the color assigned to it and that of CH_2, v. y then broadcasts an *UP* packet containing this information.

Similarly, if a node c is in 1-hop distance of a neighboring CH, i.e. CH_4, it will receive a *CA* message not intended for its own cluster. It then extracts the colors of the 1-hop nodes appearing in its CT, i.e. (CH_4) and 2-hop nodes (s and t). In such a case, an *INF* message is generated by c and the change made in its CT is included in the message (Figure 2). The *INF* message has a 1-hop list and a 2-hop list as payload containing ordered pairs of the type (*node*, *color*) which are updated with (CH_4) and (s and t) respectively.

Update (UP) message: The *UP* message is targeted at nodes within transmission range of a recently colored node, notifying the former of colors assigned to their 1 and 2-hop *neighbors*. When such a node, a, receives the *UP* message from y, CT_a is updated with the colors assigned to y, v and CH_2. An *INF*

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

message is broadcast that contains the colors and IDs of the 1-hop (y) and 2-hop (v and CH_2) neighbors of a, that were updated due to the received *UP* message (Figure 3). As 1-hop clusters are formed, the *INF* message broadcast by any node lets its CH know that its *CT* as updated. CH_3 is thus made aware through a's *INF* broadcast that the colors assigned to v, y, and CH_2 are now forbidden for a. Nodes belonging to the same cluster as the originator of the *UP* message, e.g. v, ignore it. As before, if a CH, say CH_3, receives an *UP* message from non-cluster node, say y, it implies that the node y is in 2-hop range of every member of its cluster. It adds the color of node y as a forbidden color for every node in the cluster, including a, if this information is not already present. Also, CH_3 adds the colors of all 1-hop cluster members of node y, i.e. v and CH_2, as forbidden colors for itself.

Information (INF) message: It is generated on two separate occasions. (1) The *INF* broadcast (*IB*) message is generated by a node as a result of an *UP* message is useful for its own CH and all other *neighbor* nodes. (2) An INF unicast message (*IU*), directed at the CH, is generated when a node receives an IB from a sender not belonging to the same cluster. For e.g., The IB message sent by a is used by CH_3 to update its CCT. An adjacent cluster node x, overhears the *IB*, extracts the 1-hop list (merely the color of y in this example) and updates its color table, CT_x. An *IU* message is then sent by x to CH_1 having a single node-color pair (y, *color*). We reason this as follows: The *IB* message sent by a node a contains, in the 1-hop color table, the color assigned to node y, otherwise hidden from x. For node x, y is a 2-hop neighbor and hence must appear in the 1-hop list of the incoming *IB* message. Consequently, only this information is included in the *IU* message sent by x. This two stage *INF* propagation allows CH_1 to assign a color to x such that it does not conflict with the color given to y, two hops away from x. If a node, b, receives an *INF* message from its own cluster member, a, it is ignored unless the receiving node is also the CH for that cluster. We note that the *INF* is the most commonly generated message (Figure 3): (1) The result of hearing a *CA* directly from another cluster, (2) on receiving an *UP* message and (3) another *INF* broadcast message. Successive *INF/UP* messages received at the node from different source nodes may include color information that the node has already propagated earlier. In our implementation, we adopt the following enhancement geared to reduce the number of messages in the network: The color information forwarded by a node in an INF or UP message is not sent again through another INF or UP respectively. We leave a detailed study of these scenarios for a longer version of this paper.

III. ANALYSIS

In this section we derive expressions for the expected number of channel assignment, update and inform broadcast messages for the DCA. While the analysis presented in the following propositions is approximate owing to the restricting assumptions of multi-hop modeling, all equations derived are verified in Section IV and found to be in good agreement.

Consider N nodes to be distributed independently and uniformly in a large area $R \subseteq \mathrm{R}^2$ where R^2 signifies the 2 dimensional Euclidian space. If R is large, the placement of the nodes essentially represents a Poisson process [12]. We assume:

1. All nodes have the same transmission radius r.
2. The average degree of a node is given by δ_{avg}.

Proposition 1. The total number of channel assignment messages transmitted is approximately given by,

$$n(CA) = \frac{N - CH_0}{\delta_{avg}} + CH_0 \;,\quad CH_0 = \frac{N}{\delta_{avg}+1}\left(\frac{\delta_{avg}}{\delta_{avg}+1}\right)^{\delta_{avg}}$$

Proof: For a single node A, the coverage area is πr^2. The probability that there are m nodes within its transmission range is given by: $p_m(A) = \left((\rho\pi r^2)^m / m!\right)\exp(-\rho\pi r^2)$

The expected number of nodes within its 1-hop communication range, also the average degree δ_{avg}, can be calculated as:

$$E[x] = \sum_{m=1}^{N} m p_m(A) = \sum_{m=1}^{N} m\left(\frac{(\rho\pi r^2)^m}{m!}\exp(-\rho\pi r^2)\right)$$

The weight based clustering algorithm forms 1-hop clusters by associating nodes with their 1-hop neighbors that have a higher weight, with ties being broken by node ID. If all the neighbors of a node A link with other higher weight CHs, A may be forced to announce itself CH of the single node cluster. We now calculate the expected number of such single node clusters. We list the nodes in the 1-hop neighborhood of A, along with A itself, in the following rank table. The rank of a node in this table is decided by the highest weight of its 1-hop neighbor set. From Figure 4, each neighbor of A is connected to another node having a weight higher than that of A, i.e. W1>W(A), W2>W(A) and W3>W(A) and hence A forms a 1-hop cluster. A is at position 1 on the table with the increasing index indicative of increasing rank.

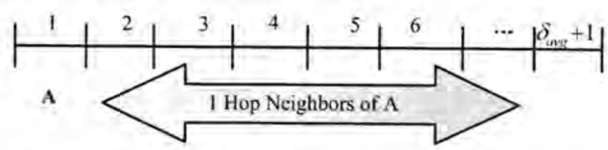

Figure 4. Ranked table of adjacent nodes on the basis of 1 hop neighbor weights.

Out of $\delta_{avg}+1$ nodes, the probability of only A occurring in slot 1 of the rank table is given by:

$$P[position1 = A] = P[A = slot1] * P[1,2,\ldots,\delta_{avg} \neq slot1]$$

$$P[position1 = A] = \frac{1}{\delta_{avg}+1}\left(1 - \frac{1}{\delta_{avg}+1}\right)^{\delta_{avg}} = \frac{1}{\delta_{avg}+1}\left(\frac{\delta_{avg}}{\delta_{avg}+1}\right)^{\delta_{avg}}$$

This is also the probability of a node forming a single node cluster as all its neighbors have been drawn away by other higher weight 2-hop neighbors. The expected number of such single node clusters is:

$$CH_0 = \frac{N}{\delta_{avg}+1}\left(\frac{\delta_{avg}}{\delta_{avg}+1}\right)^{\delta_{avg}}$$

The approximation to the average number of clusters having δ_{avg} nodes is improved by subtracting single node clusters, i.e., $(N\text{-}CH_0)/\delta_{avg}$ and it follows that the total clusters formed is approximately, $E[Clusters]=(N\text{-}CH_0)/\delta_{avg}+CH_0$. The CA

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

message is broadcast by the CH only and each cluster has a single CH. Hence *n(CA)=E[Clusters]* giving the required result.

Proposition II. The total number of update messages transmitted is approximately given by, *n(UP)=N-n(CA).*

Proof: Every node (other than the CH itself) is assigned a color by the CH of its cluster. This color assignment occurs through the *CA* message sent by the CH (or an announcement done by the node itself if it is a single node cluster). Each node of the cluster generates one *UP* message, on reception of the *CA* message. As colors are assigned only once, every non CH node will transmit a single *UP* message during the coloring process. We note that the number of CH nodes = *n(CA)*. It follows that the total number of *UP* messages equals the total number of non CH nodes and hence, *n(UP)=N-n(CA).*

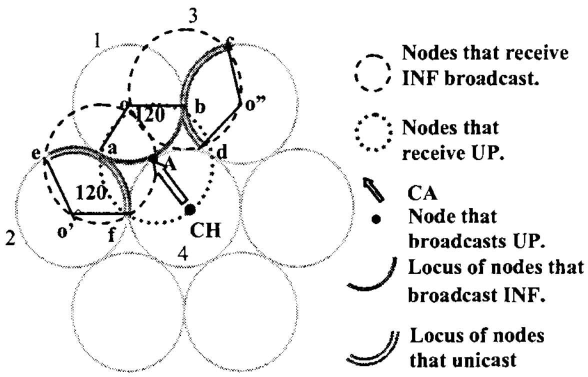

Figure 5. Calculation of *n(INF)* = *n(IB)* + *n(IU)*

Proposition III. The total number of inform messages is given by: $n(INF) = \{(2\rho/3)\pi r^2 - 1\}n(UP)$ approximately.

Proof: We prove the proposition in two parts. First we show that the average number of IB messages is given by $(\delta_{avg}/3)n(UP)$. We begin with the simplifying assumption that clusters formed are approximately of the same size (each having a radius equal to the transmission range of a node). From geometric considerations, such a cluster can have at most 6 neighboring clusters. Consider one such cluster, 4, as shown in Figure 5. The *CA* broadcast by the CH is received by node A, which then sends an *UP* message. All nodes that receive the *UP* message and send an *IB* in response, have their locus shown by sector *o-a-b* that subtends an angle of 120° to the center *o* of cluster 1. Thus the average number of nodes in this sector gives the average number of *IB* messages generated in response to a single *UP* message. For *n(UP)* messages, *n(IB)*= $\rho(120/360)\pi r^2 n(UP) = (\rho/3)\pi r^2 n(UP)$. The *IB* messages sent by the nodes in sector *o-a-b* are received by nodes lying in sectors *o"-d-c* and *o'-e-f* in the adjacent clusters 2 and 3. These are the nodes that are eligible to generate *IU* for their CHs. The number of such nodes is given by: $2\times\rho(120/360)\pi r^2 n(UP) = (2\rho/3)\pi r^2 n(UP)$. But the above set of nodes generating *IU* also includes nodes that have already heard an *UP* by the member nodes of cluster 4 that originated the *CA* message. Similarly a single node can hear more than one *IB*, and we suppress the generation of *IU* for multiple *IB* messages containing the same information. So we subtract the number of nodes that have already received *UP* and *IB* messages to get the actual number of nodes that transmit the *IU* message. This comes out as: $n(IU) = (2\rho/3)\pi r^2 n(UP) - [n(UP) + n(IB)]$. Hence the total number of *INF* messages (both *IU* and *IB*) on average is: $n(INF) = n(IU) + n(IB) = (2\rho/3)\pi r^2 n(UP) - n(UP) = \{(2\rho/3)\pi r^2 - 1\}n(UP)$.

IV. SIMULATION

In this section we perform a comparative study of the DCA and the HP-CAM as well as validate the theoretical results in section IV through computer simulations. We created a simulation environment using the JAVA based discrete event simulation platform, SimJava [13], in which we simulated the DCA and the HP-CAM with 500-1500 nodes in steps of 100, distributed randomly in a square of side 700 units. Each experimental value noted represents an average taken over 10 samples, the topology generated being similar to both the DCA and the HP-CAM. We assume a transmission radius of 40 ensuring 99% connectivity for the above densities. As metrics for comparison, we have used: *energy consumed*, *messages transmitted* and *colors required.* We neglect idle time power consumption, and associate a unit energy cost in sending and receiving a byte of information.

Results: Figure 6 compares the total number of messages transmitted for a given number of nodes. We find that the DCA performs significantly better than the HP-CAM with the difference increasing as density increases. For high densities, (1500 nodes), the DCA transmits about 40% lesser messages.

Figure 7 compares the average energy spent during the coloring process. We observe that there is a constant improvement in energy savings for all densities considered with allowed variation decided by 95% confidence intervals. We observe that despite the significant reduction in number of messages with increasing nodes, the difference between HP-CAM and DCA is approximately constant. This is primarily because the data field in each message transmitted in the HP-CAM is constant while in the DCA, the amount of information contained is a function of node degree. Increased density results in increased average degree and hence though fewer messages are sent, size of the data field is also larger. The IEEE 802.15.4 standard for sensor networks [3] defines 6 bytes in its physical layer (PHY) and 13 bytes in its MAC layer header. The energy saving incurred is significant in the DCA if the additional overhead of 19 bytes is considered for each data packet sent as seen in Figure 8.

Figure 9 shows the number of *CA*, *n(CA)*, and *UP* messages, *n(UP)* generated by the algorithm for a given node density. Analytical results derived in section IV are observed to be in good agreement with the simulation. It is seen that *n(CA)* increases at a much lower rate than *n(UP)*. This occurs primarily because with increase in node density, the average degree of a node increases. This causes each cluster to comprise of progressively larger number of nodes, causing more number of update messages to be generated for each channel assignment. At this point we note the dependence of the DCA on the clustering algorithm as *n(CA)* is dependent on the number of clusters formed and hence does not significantly

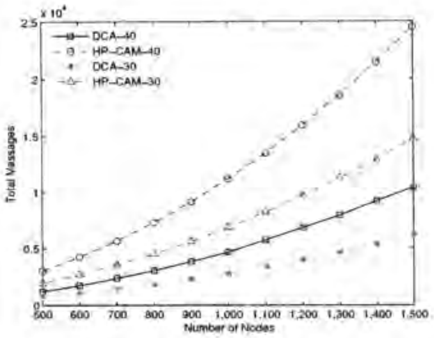

Figure6. Total number of messages generated by HP-CAM and DCA for R=30 and R =40

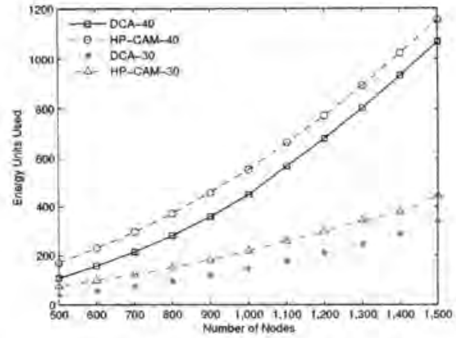

Figure7. Energy spent in HP-CAM and DCA for R=30 and R=40 for data communication only.

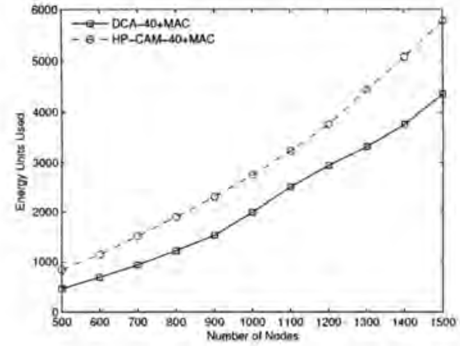

Figure8. Energy spent in HP-CAM and DCA for R=40 considering 19 bytes of PHY+MAC header.

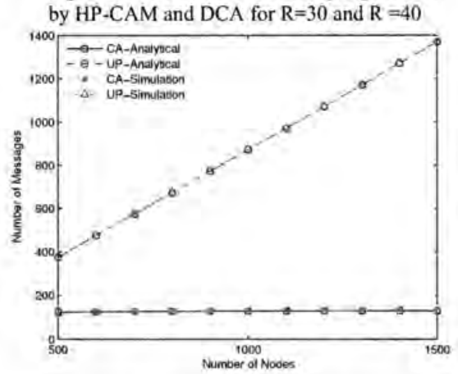

Figure9. Comparison of analytical and simulation results for CA & UP messages.

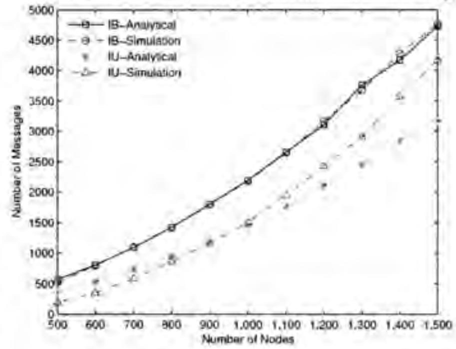

Figure 10. Comparison of analytical and simulation results for IB & IU messages.

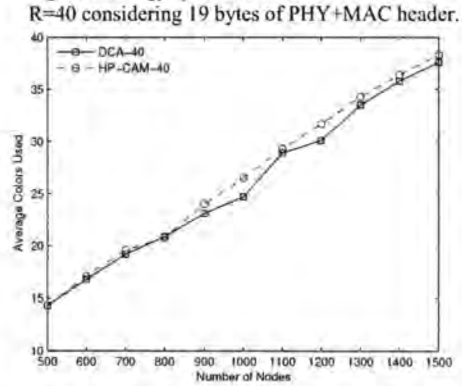

Figure11. Comparison of average colors used by DCA and HP-CAM.

change, with additional nodes preferring to join existing clusters than form separate ones by themselves. Figure 10 gives the plots for the *INF* broadcast, *n(IB)* and *INF* unicast, *n(IU)*, messages which show the expected increase with node density and in agreement with simulation and analytical values.

Figure 11 indicates the number of colors used by HP-CAM and the DCA to legally 2-hop color the network graph. We find the DCA uses approximately the same number or marginally fewer colors as the HP-CAM. This graph is important while choosing the sensor for the application. In our simulation, the IEEE 802.15.4 Zigbee (16 channels) is over the MICA2 mote while deploying 500 nodes. Similarly, the average colors predicted at a deployment of 1000 nodes ≈ 27, necessitating the choice of the WINS mote having 40 channels.

V. CONCLUSIONS

In this paper, we presented the DCA, a distributed color assignment algorithm that efficiently allocates colors to randomly deployed sensor nodes. Our proposed solution can be used as a base algorithm for several different schemes like allocating channels, assigning non-overlapping TDMA slots and nodes addresses to sensor networks. Extensive simulations as well as analysis reveals that DCA achieves a significant reduction in latency, reduced message complexity and considerable energy savings in the coloring process when compared with the existing schemes. In our future work, we plan to extend the DCA for larger k-hop clusters. Also, we plan to design and implement a MAC protocol that exploits our proposed coloring scheme to achieve higher energy efficiency without the tradeoffs of latency and throughput.

REFERENCES

[1] Website: XBOW MICA2 Mote Specifications: http://www.xbow.com/

[2] J. R. Agre, L. P. Clare, G. J. Pottie and N. P. Romanov "Development platform for self-organizing wireless sensor networks," *Aerosense, Unattended Ground Sensor Technologies and Applications*, Vol. 3713, pp. 257-268.

[3] Ed Callaway, P. Gorday, L. Hester, J.A. Gutierrez, M. Neave, B. Heile and V. Bahl, "Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks," *IEEE Communication Magazine*, Vol. 40, No. 8, pp. 70-77, August 2002.

[4] S. Ramanathan and Errol L. Lloyd, "Scheduling Algorithms for Multihop Radio Networks", *IEEE/ACM Trans. on networking*, Vol. 1, No. 2, 1993.

[5] R. Battiti, A.A. Bertossi and M.A. Bonuccelli, "Assigning code in wireless networks: bounds and scaling properties," *Wireless Networks*, Vol. 5, No. 3, August 1999, pp. 195-209.

[6] T. Herman, S. Tixeuil, "A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks," *Proceedings of the First Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors'2004)*, Finland, Lecture Notes in Computer Science, Springer-Verlag, pp. 45-58, July 2004.

[7] A.A. Bertossi and M.A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Trans. on Networking*, Vol. 3, No. 4, pp. 441-449, 1995.

[8] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Trans. on Networking*, Vol. 1, No. 6, pp. 668-677, December 1993.

[9] I. Gupta, "Minimal CDMA Recoding Strategies in Power Controlled Ad-Hoc Wireless Networks," *Technical Report*, Dept. of Computer Science, Cornell University, 2001.

[10] C. Schurgers, G. Kulkarni, and M. B. Srivastava, "Distributed on-demand address assignment in wireless sensor networks," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 13, No. 10, pp. 1056-1065, 2002.

[11] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proc. of the Intl. Symposium on Parallel Architectures, Algorithms and Networks* , pp. 310-315, June 1999.

[12] Peter Hall, *The theory of coverage processes*, John Wiley and Sons, 1988.

[13] Fred Howell and Ross McNab "Simjava: a discrete event simulation package for Java with applications in computer systems modeling," *First Intl. Conference on Web-based Modelling and Simulation*, Society for Computer Simulation, January 1998.