A Complexity Adjustable Scheduling Algorithm for Throughput Maximization in Clusterized TDMA Networks

Arash T. Toyserkani, Mohammad R. Khanzadi, Erik G. Ström, and Arne Svensson
Chalmers University of Technology
SE-412-96 Göteborg, Sweden
{arash, estrom, arnes}@chalmers.se, khanzadi@student.chalmers.se

***Abstract*—We consider clustered wireless networks, where transceivers in a cluster use a time-slotted mechanism (TDMA) to access a wireless channel that is shared among several clusters. Earlier work has demonstrated that a significant increase in network throughput can be achieved if all the schedules are optimized jointly. However, a drawback of this approach is the prohibitive level of computational complexity is required when a network with a large number of clusters and time-slots is to be scheduled. In this paper, we propose a modification to our previously proposed algorithm which allows for the complexity to be adjusted to the available processing power, provided some minimum processing power is available. This is achieved by carefully reducing the number of interfering clusters considered when scheduling a cluster. In addition, we propose and evaluate two heuristic methods of discarding the less significant clusters. While the optimality of the obtained schedule is not proven, our results demonstrate that large gains are consistently attainable.**

I. Introduction

One of the main problems in many wireless sensor networks (WSNs) today is minimizing energy consumption, the importance of which stems from the fact that in many applications, sensor nodes are small battery-powered devices. In order to conserve energy, the number of packet retransmissions in the network should be kept as low as possible. High packet-loss probability is undesirable, since it can potentially cause a high number of packet retransmissions. Another important factor in preserving energy is the duty-cycle of individual nodes. For instance, recent work on energy consumption in WSNs has shown that most wireless sensor devices consume almost as much energy when listening to the wireless channel, or even being in idle mode, as they do when actively transmitting a packet [1, Chap. 2]. From this perspective, a synchronized time-slotted medium access (MAC) scheme (TDMA) where nodes can sleep for extended periods of time seems preferable both from interference and duty-cycle points of view. For these reasons, variety of different TDMA-type MAC protocols have been proposed for implementation in WSNs [2]–[4].

In many WSN applications, the network can consist of large number of sensor nodes. In those cases, an intuitive way to reduce the transmission power and the MAC layer complexity is to locally group nodes into clusters with each cluster having a cluster-head responsible for communicating with the other clusters [1, Chap. 10]. However, having a number of clusters in close proximity with one another could potentially result in high interference and higher packet drop rate. The effect of interference can be reduced, for instance, by using spread spectrum modulation [5]. Alternatively, joint channel assignment can be used to reduce the interference caused by neighboring clusters. The channel assignment problem in clustered wireless networks is often addressed by modeling the network as directed graph [6], [7, sec. III-A-1]. This implies that all the channel gains must be known to the scheduler. Thus, this approach is not suitable in fading channels where the link gains vary over time, unless all the link gains are frequently measured and made available to the scheduler which is not applicable in WSNs.

In absence of channel measurements, statistical models of fading channel can be employed to evaluate the average performance of different schedules. One such a model is an analytical approximation of packet error rate (PER)[1] in a block interference Rayleigh fading channel as proposed in [8], [9]. In [8], the analytical model is combined with an iterative Lagrangian relaxation method to find the optimum schedule for a clusterized TDMA network and shown to result in a substantial gain in some scenarios. Nevertheless, the complexity of this algorithm increases quickly with the number of clusters and time-slots. Therefore, implementing this algorithm in a large WSNs with large number of clusters and time slots is hardly possible. To address this shortcoming, we propose a modification to the algorithm which allows for the complexity to be adjusted to the available processing power, given that at least the simplest version of the algorithm can be supported. While the optimality of the solution obtained from the simplified algorithm can not be verified, our results demonstrate that a significant gain is still attainable.

The remainder of this work is organized as follows. In Sec. II, we define the network structure and the interference model and state additional assumptions on the system. A review of the MAC algorithm proposed in [8] is given in Sec. III. The simplification of the optimum algorithm is given in Sec. IV. The performance of the proposed algorithm is analyzed and evaluated through computer simulations in sec. V, and we conclude the paper in Sec. VI.

II. System Model

Let M transceiver nodes and K data sinks be deployed over a bounded area. The nodes are indexed by integers

[1]Here, the packet error rate is assumed to be equal to the block error rate. However, in general, they are not equal but closely correlated.

$1, 2, \ldots, M$, and are clustered into K sets $\{\mathcal{C}_i\}_{i=1}^{K}$. Let a frame be an interval of time divided into W slots, indexed by $w \in \{1, \ldots, W\}$, and let $\mathcal{S}_w$ be the set of nodes, one from each cluster, scheduled for transmission in slot w. If there are fewer nodes in a cluster than the number of slots in a frame, "dummy" nodes at infinite distance from all sinks are added to the cluster. Therefore, each cluster contains exactly W nodes. In each frame, all W nodes in each cluster are to be scheduled such that (i) no more than one node from each cluster is scheduled in a given slot w, and (ii) a node can only be scheduled once per frame. A schedule $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_W\}$ that satisfies conditions (i) and (ii) is called a feasible schedule.

Each cluster is assumed to have a dedicated sink node, or cluster head, which is the receiver of all transmissions from all nodes in a cluster. The scheduling is performed by a central entity that is connected to all sinks. We assume a fixed packet length, and that all cluster heads are coarsely synchronized on a packet level, so that transmissions in a given slot takes place at approximately the same time in all clusters. It is, however, shown in [8] that gain remains significant in the presence of relatively large synchronization errors. It is worth noting that while the network architecture in this paper resembles the architecture of cellular networks, the scheduling techniques developed for cellular networks are not applicable here. This is due to the fact that in cellular networks, power and rate control are essential part of the scheduling problem. While in wireless sensor networks, the on-off power control is preferred, which results in a fundamentally different problem formulation.

Also worth mentioning that the scheduling problem defined here has a trivial solution, namely interference-free schedule, if $W \geq M$. In such a case, there are enough interference-free time slots for all the nodes. However, in practical systems this condition is not often met as such a large W also results in a large network delay and a low network throughput.

A. Interference model

The instantaneous received power from the node i at sink k is represented by $P_{i,k}$ and is defined as

$$P_{i,k} = \kappa_{i,k} \overline{P}_{i,k} \tag{1}$$

where $\overline{P}_{i,k}$ denotes the average received power from the node i at sink k and $\kappa_{i,k}$ models the effect of small-scale fading on the instantaneous received signal power. The level of mobility of nodes and the environment are assumed to be such that the small-scale fading can be modeled as block fading [10] over a single time slot. The small-scale fading is assumed to be Rayleigh distributed, hence $\kappa_{i,k}$ is a unit-mean, exponentially distributed random variable. The effects of path-loss and shadowing are captured by $\overline{P}_{i,k}$ which is assumed to be slowly varying and available to the MAC protocol either from models or measurements.

With these assumptions, the instantaneous SINR for the packet from node $\mathcal{S}_w(k) \in \{\mathcal{C}_k \cap \mathcal{S}_w\}$ to cluster head k in slot w is given by

$$\Gamma(k, \mathcal{S}_w) = \frac{\kappa_{\mathcal{S}_w(k),k} \overline{P}_{\mathcal{S}_w(k),k}}{P_{N_k} + \sum_{j \in \mathcal{S}_w, j \neq \mathcal{S}_w(k)} \kappa_{j,k} \overline{P}_{j,k}} \tag{2}$$

where P_{N_k} denotes the (known) thermal noise power at cluster head k.

B. Utility function

In this work, we use the PER model developed in [8], which provides an accurate analytical approximation of PER using a simple SINR threshold method when SINR is constant during a packet (as is the case in a TDMA system). The analytical approximation for the PER in a TDMA system in block Rayleigh fading channels, introduced in [8], is

$$P_{\text{loss}}(k, \mathcal{S}_w) = 1 - \frac{\exp\left(-\Theta \frac{P_{N_k}}{\overline{P}_{\mathcal{S}_w(k),k}}\right)}{\prod_{\substack{j \in \mathcal{S}_w, \\ j \neq \mathcal{S}_w(k)}} \left(1 + \Theta \frac{\overline{P}_{j,k}}{\overline{P}_{\mathcal{S}_w(k),k}}\right)}. \tag{3}$$

where Θ is the SINR threshold. The value of Θ only depends on the modulation format, the receiver architecture, the packet length, and the code properties. The threshold is independent of the network configuration and layout parameters such as K, M, etc. Hence, the threshold can be decided prior to network deployment, and does not need further adjustment if the network configuration changes. Due to space constraints we refer to [8], [9] for more discussion on accuracy and methods of estimating Θ.

In this work, the utility of a link from node $\mathcal{S}_w(k)$ to cluster head k, is defined as $U_k(\mathcal{S}_w) = 1 - P_{\text{loss}}(k, \mathcal{S}_w)$ and the global utility of a schedule $\mathcal{S}_w$ in slot w is given by

$$U(\mathcal{S}_w) = \sum_{k=1}^{K} U_k(\mathcal{S}_w). \tag{4}$$

Note that if $\mathcal{S}_w(k)$ is a dummy node, then $P_{\text{loss}}(k, \mathcal{S}_w) = 1$, and $U_k(\mathcal{S}_w) = 0$, i.e., dummy nodes are implicitly left out from the summation in (4). The inclusion of dummy nodes in the analysis has some interesting implications. A cluster will have dummy nodes if it has more time-slots than nodes. The optimum schedule for the dummy nodes then indicates the best time slots for radio silence in the cluster from a global network perspective.

It also worth noting that maximizing the utility function in (4) is different from maximizing the average SINR. With the utility in (4), increasing the SINR for a node beyond the point where $P_{\text{loss}} \approx 0$ does not increase the utility significantly. Conversely, the cluster utility does not change much if the SINR for a node with $P_{\text{loss}} \approx 1$ is further decreased.

III. Medium Access Control (MAC)

The purpose of this section is to briefly explain the scheduling algorithm introduced in [8]. The optimization problem can be formulated as follows.

Let $\mathcal{A}$ be a set of all feasible slot schedules i.e., $\mathcal{A} = \big\{\{c_1, \ldots, c_K\} : c_1 \in \mathcal{C}_1, \ldots, c_K \in \mathcal{C}_K\big\}$. Also let $\mathcal{A}_m$ be a set of feasible schedules where node m has been scheduled, i.e., $\mathcal{A}_m = \big\{a \in \mathcal{A} : m \in a\big\}$. The MAC problem for the K

clusters $\{\mathcal{C}_i\}_{i=1}^{K}$, and W time-slots is then

$$
\begin{aligned}
&\max_{\{\mathcal{S}_1,\ldots,\mathcal{S}_W\}} \sum_{w=1}^{W} \sum_{a\in\mathcal{A}} U(a)\,\mathrm{I}_{\mathcal{S}_w,a} \\
&\text{such that :} \\
&\text{(A)} \quad \sum_{a\in\mathcal{A}} \mathrm{I}_{\mathcal{S}_w,a} = 1,\ \forall w \in \{1,\ldots,W\}, \\
&\text{(B)} \quad \sum_{w=1}^{W} \sum_{a\in\mathcal{A}_m} \mathrm{I}_{\mathcal{S}_w,a} = 1,\ \forall m \in \{1,\ldots,M\}.
\end{aligned}
\tag{5}
$$

where $\mathrm{I}_{a,b}$ is an indicator function that is unity when $a = b$ and zero otherwise. The W constraints in (A) ensures that $\mathcal{S}_w \in \mathcal{A}$, for all $w = 1,\ldots,W$. That is, $\mathcal{S}_w$ is a feasible slot schedule. The M constraints in (B) ensures that all nodes are scheduled exactly once in a frame. Hence, (A) and (B) are satisfied if and only if $\{\mathcal{S}_1,\ldots,\mathcal{S}_W\}$ is a feasible schedule.

A brute-force solution to (5) can be obtained by evaluating $\sum_{w=1}^{W} U(\mathcal{S}_w)$ for all possible schedules and choose the optimum one. It is easy to show that there are as many as $(W!)^K$ different feasible schedules. Therefore, the complexity of a brute-force solution to (5) quickly becomes prohibitive as the number of clusters and time slots in the network grows. It is, however, shown in [8] that a near optimum solution to (5) can be obtained using an iterative Lagrangian relaxation method with considerably less computational complexity. A brief description of this algorithm is as follows. The algorithm forms a feasible candidate schedule by first fixing the schedule for cluster 1, and then for cluster $r = 2, 3, ..., K$ in succession. When considering the rth cluster, (A) and (B) are relaxed such that (5) becomes a two-dimensional assignment problem. The utility for all possible combinations of the unscheduled nodes, i.e., the nodes from cluster $r, r+1, ..., K$, must be computed. The possible combinations can be represented as the leaves of a W-ary rooted tree of depth $K - r + 1$, which we will call the node tree. These utilities are feed into the auction algorithm which can efficiently solve this two-dimensional assignment problem [11]. The obtained schedule is fixed and algorithm proceed to the next cluster. The flowchart of the algorithm is given in Fig. 1. Interested readers are referred to [8] and references therein for more detailed description of the algorithm.

IV. Complexity reduction

Most of the complexity in the scheduling algorithm described previously is due to the generation of the node tree and the subsequent discovery of the best possible utilities for each time slot. When scheduling the cluster r, clusters 1 to $r - 1$ are already scheduled and therefore, they have fix schedules (see Fig. 1). Thus, the node tree of a time slot at cluster r contains all the possible combinations of nodes from clusters r to K and fixed nodes from clusters 1 to $r - 1$.

Excluding the first cluster, finding the best utility for every node in cluster r and at a given time slot requires a search over a tree with W^{K-r+1} leaves. While this search is considerably less complex than the brute force approach (which searches $W!^K$ feasible schedules), the complexity of this search also

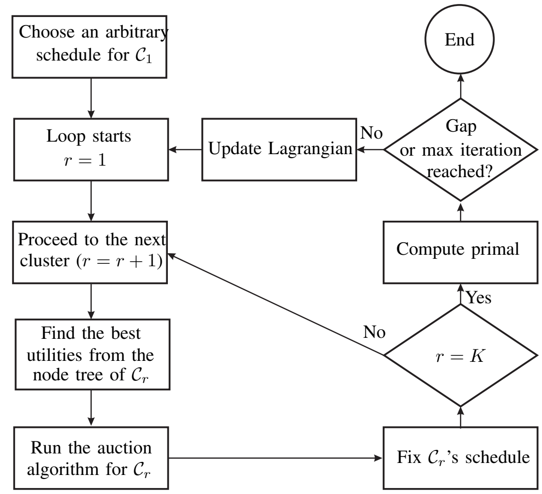

Fig. 1. The flowchart of the algorithm in [8].

becomes quickly prohibitive as the number of clusters and time slots are increased.

In addition, storing this large node tree can occupy a large amount of memory. For instance, assuming that each branch of the tree occupies 2 bytes in memory, a network of 16 cluster and 8 time slot requires more than 70 Terabyte of memory. Regardless of the details of the memory handling techniques, the complexity of this scheduling algorithm for a large number of clusters and time slots is dominated by the complexity embedded in generating, storing, and searching this large node tree. Our aim in this paper is to reduce this complexity such that it can be adjusted to the available processing power.

Lets assume that for a given available processing power and a given number of time-slots per frame (i.e., W), a tree with maximum W^{K_c+1} leaves can be processed, where $K_c \leq K$. Thus, for any cluster r where $r \geq K - K_c$, the scheduling can be done with no simplifications. However, if $r < K - K_c$, the tree size can not be handled by the available processing power and therefore, the tree size must be reduced. One way to prune the node tree is to consider only K_c clusters out of the $K - r$ unscheduled clusters in the node tree. Our approach is motivated by the fact that clusters contributions to the scheduling gain are not equal. Thus, if the less significant clusters can be identified and removed from the tree, the complexity can be reduced with little loss in the performance. To identify these clusters, we define a metric $\eta_{i,r}$ which represents the relevance of cluster i for the scheduling of cluster r. Two heuristic methods for calculating this metric, namely *AvePow* and *VarPow*, are presented in Sec. IV-A and IV-B respectively. The modified flowchart of the algorithm that takes these changes into account is shown in Fig. 2.

However, if clusters are removed from the node tree, the dual is no longer valid and the duality gap can not be obtained. Therefore, the optimality of the found solution can not be verified. In addition, the stop criterion of the algorithm can no longer be based on the duality gap. Nevertheless, our results presented in Sec. V demonstrate that if the discarded clusters are chosen wisely, a significant performance gain is still possible.

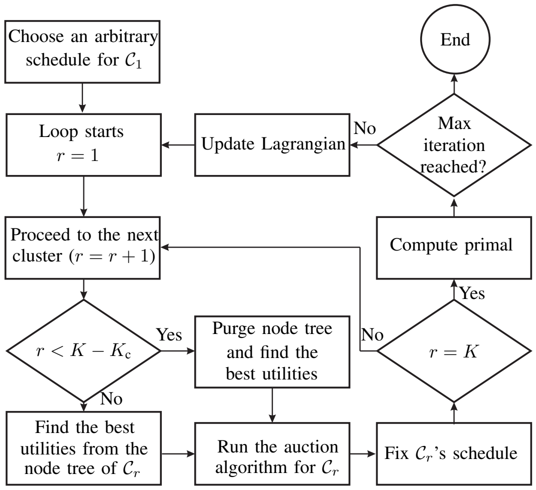

Fig. 2. The flowchart of the simplified scheduling algorithm.

A. AvePow Heuristic

It is shown in [8] that in a low inter-cluster interference environment, there is little to be gained from cooperatively scheduling clusters. Exploiting this observation, $\eta_{i,r}$ is defined here as the the average interfering power of nodes in cluster i to the sink of cluster r. Formally

$$\eta_{i,r} = \frac{1}{W} \sum_{j \in \mathcal{C}_i}^{W} \overline{P}_{j,r} \qquad i = r+1, r+2, ..., K \tag{6}$$

and the pruned node tree is made up from the rth cluster and the K_c clusters with the highest metrics.

B. VarPow Heuristic

It is also shown in [8] that in addition to the interference level of a cluster, the variations of the interfering power of nodes inside a cluster also effects the amount of gain obtained by the scheduling algorithm. For instance, consider a cluster that nodes are closely packed together (i.e. very similar interference level at sink of cluster r). In this case, any combination of nodes in this cluster yield similar utilities for cluster r. Therefore, there is little loss in utility if this cluster is ignored regardless of the level of interfering power. Therefore, in the VarPow method, the average interfering power of nodes from a cluster is scaled with the standard deviation of the average interference powers. To simplify the notation, let $\mu_{i,r} = \frac{1}{W} \sum_{j \in \mathcal{C}_i}^{W} \overline{P}_{j,r}$. Then we have

$$\eta_{i,r} = \mu_{i,r} \sqrt{\frac{1}{W} \sum_{j \in \mathcal{C}_i} (\overline{P}_{j,r} - \mu_{i,k})^2} \qquad i = r+1, r+2, ..., K \tag{7}$$

As a result, the importance of clusters with small interference variations are reduced while the clusters with large interference variations gain in importance.

V. Numerical Analysis and Discussion

A. System set-up

We consider a short-range clustered WSN, where all nodes transmit packets of 100 bytes each at a fixed output power P, modulated with BPSK signaling and with no error correction. The receiver is a matched filter followed by a hard-decision detector. For this system, Θ is found to be 3.03. To simplify our simulations, shadow fading is ignored and the average received power $\overline{P}_{i,k}$ is modelled by the log-distance path-loss model as follow

$$\overline{P}_{i,k} = P_0 \left(\frac{d_0}{d_{i,k}} \right)^{\alpha}$$

where $d_{i,k}$ is the distance between node i and sink k and P_0 is the average received power at distance d_0. In the simulations, $P_0/P_N = 10$ dB at reference distance $d_0 = 1$ m and the path-loss exponent is $\alpha = 3$. The results presented here are for a network of 16 clusters where sinks are located in a 4×4 grid at regular horizontal and vertical distance of $R = 1$ m (see Fig. 3). Each frame is assumed to be 8 time slot and each cluster has 4 nodes and 4 dummy nodes (i.e. 4 empty time-slots).

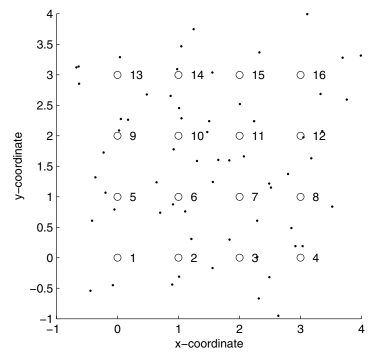

Fig. 3. An example of the simulated network of 16 clusters with sinks (circles) and nodes (dots).

To emulate a network configuration where a clustering algorithm, e.g., LEACH [5], has been executed, the node coordinates in the a cluster is randomly generated as realizations of a circular Gaussian distribution with mean equal to the coordinates of the sink and standard deviation σ_R. In this work, σ_R is assumed to be $R/2$. The distance between sinks, σ_R, α, and P_0/P_N together determines the expected SINR conditions in the network.

As mentioned in Sec. IV, after applying the complexity reduction method, the dual is not valid anymore. Thus, the iterative algorithm can not be terminated using the duality gap and must be stopped after a fixed number of iterations. To examine the convergence of the algorithm as the function of number of iterations, a series of simulations were performed with 10 iteration each. In each simulation, the iteration number of the best performing schedule is recorded. One such result for $K_c = 2$ and AvePow method is shown in Fig. 4. Similar behavior are observed from other simulations. As it can be observed from this figure, about 90% of the time, the best schedule is found in the first 5 iterations. Similar results are also reported in [8]. As a result, in the rest of this paper, the algorithm is set up to terminate after 5 iterations.

It is also of interest to investigate the effect of K_c on the achieved utility. To observe this, simulations were performed for AvePow method and for $K_c = 1$ and 3 and the resulting

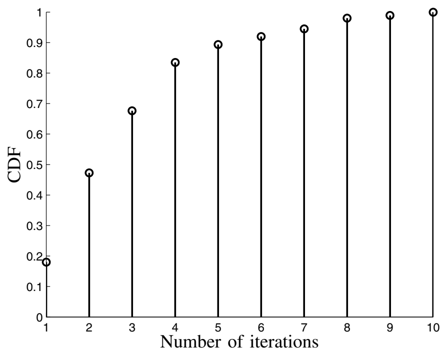

Fig. 4. The cumulative distribution function (CDF) of the iteration number for which the best performing schedule is found.

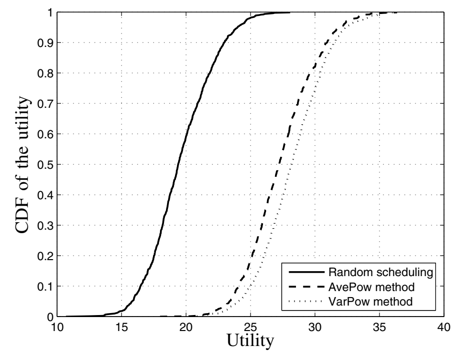

Fig. 6. The CDF of utility for AvePow versus VarPow for $K_{\mathrm{c}} = 1$.

utilities are reported in Fig. 5. Results presented in this figure suggest that the majority of gain may be attainable by only considering few neighboring clusters. It worth noting, however, that the amount of gain for higher K_{c} is highly dependent of topology. The higher is the number of clusters with many neighboring clusters, the more gain is expected from higher K_{c} values. The interference model may also effect the results.

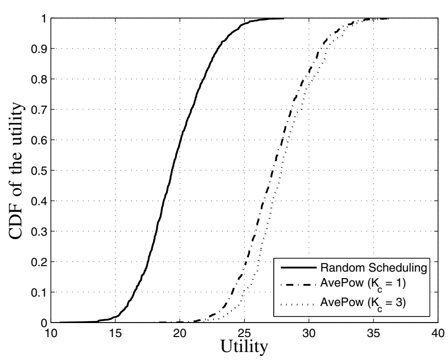

Fig. 5. The CDF of utility for AvePow method and for $K_{\mathrm{c}} = 1$ and 3 versus random scheduling utility.

Additional simulations were performed to compare the performance of AvePow and VarPow methods and results are presented in 6. As it can be seen, when the power variations in the cluster is taken into account, i.e. VarPow, higher utility gain is possible. The topology and the interference model again may significantly effect the amount of achievable gain.

VI. Conclusions

In this paper, we outline a method for reducing the complexity of the iterative Lagrangian scheduling algorithm [8]. The proposed approach intelligently reduces the size of the network from the perspective of a given cluster such that the complexity of the scheduling algorithm is limited to the available processing power. Two heuristic methods to regulate the size reduction, namely AvePow and VarPow, are also proposed. While the optimality of the algorithm can not be verified, the simulation results demonstrate a significant gain over the random scheduling.

Our results also show that in the simulated scenarios in this paper, the best performing schedule is found in the first 5 iterations in about 90% of simulations. In addition, our results show that considering as few as 3 neighboring clusters offer the majority of the gain. These results indicate that the method introduced here provides a significant complexity reduction with a little loss in the performance. However, it worth noting that the exact amount of gain and other numerical values provided here depend largely on topology details and the interference model. Therefore, these values may vary if a different topology or an interference model are considered.

References

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2006.

[2] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *Wireless Personal Communications*, vol. 7, no. 5, 2000.

[3] M. Gerla and J. Tzu-Chieh Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[4] J. Li and G. Y. Lazarou, "A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks," in *Third International Symposium on Information Processing in Sensor Networks, IPSN 2004*, 2004, pp. 55–60.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, 2002.

[6] S. Ramanathan, "Unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.

[7] J. A. Stankovic, T. F. Abdelzaher, C. Lu, L. Sha, and J. C. Hou, "Real-time communication and coordination in embedded sensor networks," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1002–1022, 2003.

[8] A. T. Toyserkani, M. Rydström, E. G. Ström, and A. Svensson, "A scheduling algorithm for minimizing the packet error probability in clusterized TDMA networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1–11, 2009.

[9] A. T. Toyserkani, E. G. Ström, and A. Svensson, "A semi-analytical approximation to the block error rate in Nakagami-*m* block fading channels," Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, Tech. Rep. R011/2009, 2009.

[10] R. J. McEliece and W. E. Stark, "Channels with block interference," *IEEE Transactions on Information Theory*, vol. IT-30, no. 1, pp. 44–53, 1984.

[11] D. P. Bertsekas, "The auction algorithm for assignment and other network flow problems: a tutorial," *Interfaces*, vol. 20, pp. 133–149, 1990.