CHALMERS
UNIVERSITY OF TECHNOLOGY

A Scheduling Algorithm for Minimizing the Packet Error Probability in Clusterized TDMA Networks

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 804621, 10 pages
doi:10.1155/2009/804621

Research Article

A Scheduling Algorithm for Minimizing the Packet Error Probability in Clusterized TDMA Networks

Arash T. Toyserkani, Mats Rydström, Erik G. Ström, and Arne Svensson

Department of Signals and Systems, Chalmers University of Technology, 412-96 Göteborg, Sweden

Correspondence should be addressed to Arash T. Toyserkani, arash@chalmers.se

Received 1 December 2008; Revised 18 April 2009; Accepted 25 July 2009

Recommended by Wing-Kin Ma

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. An approximate expression for the packet-loss probability is derived for networks with one or more mutually interfering clusters in Rayleigh fading environments, and the approximation is shown to be good for relevant scenarios. We then present a scheduling algorithm, based on Lagrangian duality, that exploits the derived packet-loss model in an attempt to minimize the average packet-loss probability in the network. Computer simulations of the proposed scheduling algorithm show that a significant increase in network throughput can be achieved compared to uncoordinated scheduling. Empirical trials also indicate that the proposed optimization algorithm almost always converges to an optimal schedule with a reasonable number of iterations. Thus, the proposed algorithm can also be used for bench-marking suboptimal scheduling algorithms.

1. Introduction

One of the problems with many wireless networks today is energy consumption, stemming from the fact that modern radio transceivers are often battery powered, and, hence, energy is a scarce resource that needs to be conserved as much as possible. Complexity is another important issue, since, many wireless network applications require the size and cost of individual network nodes to be kept at a minimum. One important example of the above is wireless sensor networks (WSNs) [1, Chapter 1], that have lately received considerable attention, both from industry and academia.

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, Chapter 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. However, interference will still be present if two or more networks, or "clusters" of nodes, are colocated in close vicinity of each other.

In the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol [2], a TDMA-type MAC scheme tailored for WSNs is integrated with clustering and routing mechanisms. In LEACH, each cluster chooses a random spreading sequence that is used locally. This reduces intercluster interference but also increases the complexity in each node. Another WSN protocol that uses a TDMA-type MAC-scheme is the Self-Organizing Medium Access Control for Sensor Networks (SMACS) protocol [3]. SMACS implements both distributed neighborhood discovery and TDMA scheduling. In SMACS, all nodes are assumed to know the time duration of a so-called "superframe". Many other TDMA-based MAC mechanisms have been also proposed for implementation in clusterized networks [4, 5].

One frequently occurring drawback with MAC design proposals is that overly simplistic propagation models are used, for example, not accounting for Rayleigh fading effects. For instance, channel assignment problem in wireless network is often addressed by modelling the network as directed graph [6], [7, Section III-A-1], and [8]. This assumption is not suitable in fading channels where the link gains vary over time, unless all the instantaneous link gains are frequently measured and made available to the scheduler, resulting in much added overhead and complexity.

In this work, we include Rayleigh fading and log-distance path loss in the system model and propose a TDMA-type MAC mechanism that jointly schedules transmissions between nodes and cluster heads with the objective to minimize the average packet error rate (PER), (The packet error rate is assumed to be equal to the block error rate. However, in general, they are not equal but closely related.) that is, to maximize the total network throughput. We make the assumption that all nodes have a fixed output transmission power and formulate the scheduling problem as an integer programming problem, more specifically an assignment problem.

Similar approaches are taken in [9, 10], where joint opportunistic power scheduling and rate control problems are considered. Our work differs from [9, 10], and references cited therein, on three main points (a) Instead of allowing a smooth tuning of transmitter output powers, we impose an on-off constraint on transmitters. One of the main reasons is that power consumption is sometimes only weakly correlated with transmit power [1, Chapter 2]. (b) Instead of the signal to interference and noise ratio (SINR), we consider the PER (a nonlinear function of SINR) to be the main optimization objective. While the PER is a more relevant measure, the SINR is often preferred in the literature due to the lack of a tractable analytical solution for the PER for a wide range of different modulations, coding methods, and fading channels [11]. To overcome this, a closed-form formula for estimation of the PER in block faded Rayleigh channels in presence of interference is derived and shown to be highly accurate. Finally, (c) in order to make the sleep time as long and uninterrupted as possible, we do not schedule nodes on a slot-by-slot basis. Instead, we schedule all slots in a frame in one run of the algorithm such that no node receives more than one slot.

The remainder of this paper is organized as follows. In Section 2, we define the network and interference model and state additional assumptions on the system. The utility function based on our analytical approximation of PER is introduced in Section 3. The interference model is later used in the proposed MAC algorithm, that is derived in Section 4. The proposed algorithm is analyzed and evaluated through computer simulation in Section 5, and we conclude the paper in Section 6.

2. System Model

Let M transceiver nodes and K data sinks be deployed over a bounded area. The nodes are indexed by integers $1, 2, \ldots, M$, 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. It is worth noting that a very large W may result in a trivial interference-free schedule where in each time slot only one real node is scheduled with dummy nodes from all other clusters. However, setting W arbitrarily large is not possible in a majority of practical systems as it also results in a large network delay and a low network throughput. The problem of how to adjust W and how to select a subset of nodes when the number of nodes per cluster is larger than W is not considered here.

Based on these assumptions, each cluster contains exactly W nodes. In each frame, all W nodes in each cluster are to be scheduled such that no more than one node from each cluster is scheduled in a given slot w, and a node can only be scheduled once per frame. A schedule $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_W\}$ that satisfies these conditions is called a feasible schedule. Each cluster is assumed to have a dedicated sink node, or cluster head. Similar to a Bluetooth system [1, Chapter 5], the cluster head 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. While these settings resemble a cellular network architecture, 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.

We assume the packet length is fixed, 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. However, synchronization errors among clusters are considered in the numerical evaluations of the proposed algorithm (Section 5).

2.1. 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 modelled as block fading [12] 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.

3. Utility Function

It is shown in [11] that in an interference free environment, the PER of block coded packets in block faded Rayleigh channels can be accurately approximated by a simple SNR threshold. That is each received packet is considered to be successful if the instantaneous SNR is above a given threshold Θ, and lost otherwise. Hence, the PER in the block faded Rayleigh channels is approximated by [11]

$$P_{\text{loss}}\left(\overline{\Gamma}\right) = \Pr\{\Gamma < \Theta\} = 1 - \exp\left(\frac{\Theta}{\overline{\Gamma}}\right), \tag{3}$$

where $P_{\text{loss}}(\overline{\Gamma})$ is the PER estimate based on SNR threshold model and $\overline{\Gamma}$ is the average SNR. Similar results for turbo coded packets are reported in [13].

In this section, we examine if applying a similar method in presence of interference results in an accurate approximation of PER. In TDMA systems with block fading channels, SINR is constant during one time slot (if the intercluster synchronization error is small). Therefore, applying the threshold method results in

$$\begin{aligned} P_{\text{loss}}(k, \mathcal{S}_w) &= \Pr\{\Gamma(k, \mathcal{S}_w) < \Theta\} \\ &= \Pr\left\{\kappa_{\mathcal{S}_w(k),k} < \Theta\frac{P_{N_k}}{\overline{P}_{\mathcal{S}_w(k),k}} + \sum_{\substack{j\in\mathcal{S}_w,\\ j\neq\mathcal{S}_w(k)}} \kappa_{j,k}\Theta\frac{\overline{P}_{j,k}}{\overline{P}_{\mathcal{S}_w(k),k}}\right\}. \end{aligned} \tag{4}$$

Since all fading coefficients are i.i.d. unit-mean exponential random variables, we have, as shown in the appendix,

$$P_{\text{loss}}(k, \mathcal{S}_w) = 1 - \frac{\exp\left(-\Theta\left(P_{N_k}/\overline{P}_{\mathcal{S}_w(k),k}\right)\right)}{\prod_{j\in\mathcal{S}_w, j\neq\mathcal{S}_w(k)}\left(1 + \Theta\left(\overline{P}_{j,k}/\overline{P}_{\mathcal{S}_w(k),k}\right)\right)}. \tag{5}$$

The accuracy of this model is verified by comparing the PER for the node-sink link of node m, denoted by $\overline{P}_{e,m}$, with P_{loss} for the same link. An analytical expression for $\overline{P}_{e,m}$ can be obtained by integrating the instantaneous PER over the SINR variations where the instantaneous PER for node $m = \mathcal{S}_w(k)$ is given by

$$P_{e,S_w(k)} = 1 - \sum_{i=0}^{t}\binom{n}{i} p(\Gamma(k, S_w))^i [1 - p(\Gamma(k, S_w))]^{n-i}, \tag{6}$$

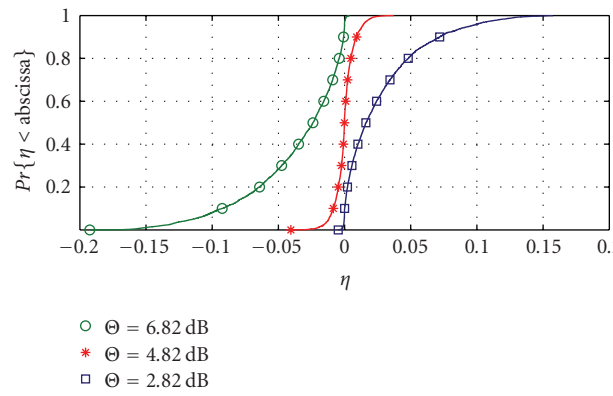

Figure 1: CDF of capture model error, $\eta = \overline{P}_{e,m} - P_{\text{loss}}$.

where $p(\Gamma)$ is the bit error rate at the given Γ and t is the error correction capability (in number of bit errors). Interested readers are referred to [14] and references cited therein, for more information regarding the block error probability of various coding and decoding methods.

Since the closed-form analytical solution to $\overline{P}_{e,m}$ is untractable [11], Monte-Carlo simulation is used in this paper to estimate $\overline{P}_{e,m}$. The required statistics were obtained through simulations of 200 randomly generated networks. For each the node-sink link, $\overline{P}_{e,m}$ is obtained by averaging over 1000 Rayleigh fading realizations. The other simulation parameters can be found in Section 5.

In Figure 1, we plot the cumulative distribution function (CDF) of the difference $\eta = \overline{P}_{e,m} - P_{\text{loss}}$ evaluated for all links in all networks. We see that, for a correctly chosen threshold, in this case $\Theta = 4.82$ dB, the error is quite small. We also note that variations as large as ± 2 dB in the threshold increases the error, but not significantly, and hence the packet-loss model is not overly sensitive to the choice of threshold.

Finally, we note that the choice of Θ only depends on the modulation format, that is, BPSK, the receiver architecture, the packet length, and the properties of the code. The threshold does not depend on the network configuration and layout, that is, K, M, and so forth. Hence the threshold can be decided prior to network deployment, and does not need to be reconfigured if the network configuration changes. For methods of finding Θ, interested readers are referred to [11].

To isolate the effect of the proposed P_{loss} formula, the reliability, or "utility", of a link from node $\mathcal{S}_w(k)$ to cluster head k, is defined as $U_k(\mathcal{S}_w) = 1 - P_{\text{loss}}(k, \mathcal{S}_w)$. Adding more terms to $U_k(\mathcal{S}_w)$ does not change the optimization algorithm in Section 4 as long as utility of each schedule can be obtained independently of other schedules.

The global utility of a schedule $\mathcal{S}_w$ in slot w is then given by

$$U(\mathcal{S}_w) = \sum_{k=1}^{K} U_k(\mathcal{S}_w). \tag{7}$$

Note that if $\mathcal{S}_w(k)$ is a dummy node, then $P_{\text{loss}}(k, \mathcal{S}_w) = 1$, and $U_k(\mathcal{S}_w) = 0$, that is, dummy nodes are implicitly left

out from the summation in (7). 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 schedule for the dummy nodes then indicates the best time slots for radio silence in the cluster from a global network perspective.

The utility function $U(\mathcal{S}_w)$ in (7) does not necessarily need to consider all clusters. The throughput and its subsequent optimization from a subset of clusters' point of view are obtained by simply removing appropriate terms from the sum in (7). The implications of this are analyzed in Section 5. We also emphasize that maximizing the utility function in (7) is different from maximizing the average SINR. With the utility in (7), 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.

4. Medium Access Control

The aim of the proposed Medium Access Control (MAC) layer is to schedule node transmissions such that the average probability of a successful packet delivery in the network is maximized. Due to the assumed slotted MAC scheme, this will also maximize the network throughput. We define $\mathcal{A}$ as a set of all feasible slot schedules, that is, $\mathcal{A} = \{\{c_1, \ldots, c_K\} : c_1 \in \mathcal{C}_1, \ldots, c_K \in \mathcal{C}_K\}$. We also define $\mathcal{A}_m$ as a set of feasible schedules where node m has been scheduled, that is, $\mathcal{A}_m = \{a \in \mathcal{A} : m \in a\}$.

The MAC problem for the K clusters $\{\mathcal{C}_i\}_{i=1}^K$ and W time slots is then

$$\max_{\{\mathcal{S}_1,\ldots,\mathcal{S}_W\}} \sum_{w=1}^{W} \sum_{a\in\mathcal{A}} U(a)\mathbf{I}_{\mathcal{S}_w,a}$$

such that

$$\text{(A)} \quad \sum_{a\in\mathcal{A}} \mathbf{I}_{\mathcal{S}_w,a} = 1, \quad \forall w \in \{1,\ldots,W\}, \qquad (8)$$

$$\text{(B)} \quad \sum_{w=1}^{W} \sum_{a\in\mathcal{A}_m} \mathbf{I}_{\mathcal{S}_w,a} = 1, \quad \forall m \in \{1,\ldots,M\}.$$

Here, and throughout the rest of this work, $\mathbf{I}_{a,b}$ 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) ensure that all nodes are scheduled in exactly one slot. Hence, (A) and (B) are satisfied if and only if $\{\mathcal{S}_1,\ldots,\mathcal{S}_W\}$ is a feasible schedule.

As the number of nodes and clusters in the network grows, the complexity of a brute-force solution to (8) quickly becomes prohibitive. In fact, there are as many as $(W!)^K$ different feasible schedules to choose from.

4.1. MAC Problem for Two Clusters. Since we assume no time dependence, the utility function $U(\mathcal{S}_w)$ only depends on the coscheduled nodes in the slot schedule $\mathcal{S}_w$ and not on the specific slot w. Hence, a permutation of slot schedules in a global schedule $\{\mathcal{S}_1,\ldots,\mathcal{S}_W\}$ will not affect the utility $\sum_{w=1}^{W} U(\mathcal{S}_w)$. We can therefore arbitrarily choose any feasible schedule for nodes in, for example, cluster $\mathcal{C}_1$, without loss in maximum achievable utility. After fixing the schedule $\{\mathcal{S}_w(1)\}_{w=1}^W$ for nodes in $\mathcal{C}_1$ in a two-cluster network, the MAC problem in (8) reduces to the two-dimensional assignment problem:

$$\max_{\{\mathcal{S}_1(2),\ldots,\mathcal{S}_W(2)\}} \sum_{w=1}^{W} \sum_{c_2\in\mathcal{C}_2} U(\{\mathcal{S}_w(1), c_2\})\mathbf{I}_{\mathcal{S}_w(2),c_2}$$

such that

$$\text{(A)} \quad \sum_{c_2\in\mathcal{C}_2} \mathbf{I}_{\mathcal{S}_w(2),c_2} = 1, \quad \forall w \in \{1,\ldots,W\}, \qquad (9)$$

$$\text{(B}_2\text{)} \quad \sum_{w=1}^{W} \mathbf{I}_{\mathcal{S}_w(2),c_2} = 1, \quad \forall c_2 \in \mathcal{C}_2.$$

Unlike the case of a multidimensional assignment problem, efficient algorithms exist that solve (9) in polynomial time such as maximum weight matching problem on bipartite graph [15]. We use the auction algorithm, due to Bertsekas [16], [17, Chapter 6], briefly described below.

Consider problem (9), where the schedule for nodes in $\mathcal{C}_1$, that is, $\{\mathcal{S}_w(1)\}_{w=1}^W$, is fixed and known. The auction algorithm for solving this problem is as follows. (a) Envision the nodes in $\mathcal{C}_2$ as objects on sale at an auction, and envision the slots as buyers at the auction. Initially, the asking prices $\{p_{c_2}\}_{c_2\in\mathcal{C}_2}$ of the objects on sale are set to zero. (b) Let each slot w successively "place a bid" on the node $i = \arg\max_{c_2\in\mathcal{C}_2}\{U(\{S_w(1), c_2\}) - p_{c_2}\}$, that is, the node that yields the highest net value $v_i = U(\{S_w(1), i\}) - p_i$ for slot w. (c) When a node is bid upon, its asking price p_i is raised by $v_i - z_i + \beta$, where $\beta > 0$ is a small number, and $z_i = \max_{c_2\in\mathcal{C}_2, c_2\neq i}\{U(\{S_w(1), c_2\}) - p_{c_2}\}$, that is, the second best net value for slot w. The reason for the additional small increase in price, β, is to prevent ties among buyers, a topic further discussed in [17]. In this paper we let $\beta = 1/W$. (d) The auction continues until all nodes have received at least one bid, at which point a solution to (9) has been found.

Although of little consequence for the development here, it is interesting to note that the auction algorithm is actually a dual method on its own. It can be shown that when the auction algorithm terminates, the node asking prices solves

$$\min_{\{p_{c_2}\}_{c_2\in\mathcal{C}_2}} \sum_{w=1}^{W} \sup_{c_2\in\mathcal{C}_2} \left(U(\{\mathcal{S}_w(1), c_2\}) - p_{c_2}\right) + \sum_{c_2\in\mathcal{C}_2} p_{c_2}, \qquad (10)$$

to within any $\epsilon > 0$ of the optimal value (ϵ depends on the choice of β). This is, in fact, the dual problem to (9), after relaxation of constraints on nodes in $\mathcal{C}_2$.

For a complete derivation, additional discussions, and results on the auction algorithm, the reader is referred to [16], [17, Chapter 6], and references cited therein.

4.2. MAC Problem for Arbitrary Number of Clusters. It was noted above that the complexity of a brute-force solution to (8) grows quickly with W and K. However, if a relaxed problem, that is, the maximization of a Lagrangian, can be easily solved, and we also have access to a good method that converts a solution to the relaxed problem into one that is primal feasible, then experience with similar types of combinatorial optimization problems; see, for example, [9, 10, 17–19], and references cited therein, gives that an iterative solution of the dual problem often yields a near optimal, or even an optimal solution to the primal problem. Hence, the algorithm we propose is an iterative algorithm similar to one in [18], where each iteration involves the following three steps. (1) Given a vector of dual variables, a relaxed version of (8) is solved. (2) A primal feasible schedule is constructed from the solution to the relaxed problem and the vector of dual variables. (3) If the obtained primal solution is found to be unsatisfactory, then the dual variables are updated, and we iterate again.

4.2.1. The Relaxation Step. We relax constraints on nodes in $\mathcal{C}_3, \mathcal{C}_4, \ldots, \mathcal{C}_K$ in (8). Let μ_{c_p} denote the dual variable associated with node $c_p \in \mathcal{C}_p$. The dual function is then

$$q(\boldsymbol{\mu}) = \sum_{c_3 \in \mathcal{C}_3} \mu_{c_3} + \cdots + \sum_{c_K \in \mathcal{C}_K} \mu_{c_K} + \sup_{\{\mathcal{S}_1, \ldots, \mathcal{S}_W\}} \sum_{w=1}^{W} \sum_{c_1 \in \mathcal{C}_1} \cdots \sum_{c_K \in \mathcal{C}_K} \left(U(\{c_1, \ldots, c_K\}) - \mu_{c_3} \cdots - \mu_{c_K}\right) \mathbf{I}_{\mathcal{S}_w, \{c_1, \ldots, c_K\}}$$

such that

$$\begin{aligned} &(\mathrm{A}) \quad \sum_{a \in \mathcal{A}} \mathbf{I}_{\mathcal{S}_w, a} = 1, \quad \forall w \in \{1, \ldots, W\}, \\ &(\mathrm{B}_{12}) \quad \sum_{w=1}^{W} \sum_{a \in \mathcal{A}_m} \mathbf{I}_{\mathcal{S}_w, a} = 1, \quad \forall m \in \{\mathcal{C}_1 \cup \mathcal{C}_2\}, \end{aligned} \tag{11}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{(K-2)W}$ contains all dual variables. As for the case of $K = 2$ in Section 4.1, we can use any feasible schedule for the nodes in, for example, $\mathcal{C}_1$, without loss in maximum achievable utility. Let $V^{(2)}(w, c_2) = \sup_{c_3 \in \mathcal{C}_3, \ldots, c_K \in \mathcal{C}_K} \{U(\{\mathcal{S}_w(1), c_2, \ldots, c_K\}) - \mu_{c_3} \cdots - \mu_{c_K}\}$, then the problem in (11) is equivalent to

$$q(\boldsymbol{\mu}) = \sum_{c_3 \in \mathcal{C}_3} \mu_{c_3} + \cdots + \sum_{c_K \in \mathcal{C}_K} \mu_{c_K} + \sup_{\{\mathcal{S}_w(2)\}_{w=1}^{W}} \sum_{w=1}^{W} \sum_{c_2 \in \mathcal{C}_2} V^{(2)}(w, c_2) \mathbf{I}_{\mathcal{S}_w(2), c_2}$$

such that

$$\begin{aligned} &(\mathrm{A}) \quad \sum_{c_2 \in \mathcal{C}_2} \mathbf{I}_{\mathcal{S}_w(2), c_2} = 1, \quad \forall w \in \{1, \ldots, W\}, \\ &(\mathrm{B}_2) \quad \sum_{w=1}^{W} \mathbf{I}_{\mathcal{S}_w(2), c_2} = 1, \quad \forall c_2 \in \mathcal{C}_2. \end{aligned} \tag{12}$$

Hence, for a given vector of dual variables $\boldsymbol{\mu}$, this problem is a two-dimensional assignment problem which is easily solved, as was shown in Section 4.1. We note that, to compute $V^{(2)}(w, c_2)$, a search over W^{K-2} slot assignments is necessary. In the scenarios considered in this work, an exhaustive search is feasible. However, larger networks may require the addition of more advanced search methods, such as branch and bound techniques, further discussed in Section 5.

4.2.2. A Method for Generating Feasible Schedules. When solving (12), we implicitly obtain a feasible scheduling of nodes from clusters $\mathcal{C}_1$ and $\mathcal{C}_2$. A feasible schedule that also includes nodes from remaining clusters must now be generated. In general, a schedule that is feasible for nodes in $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{r-1}$, for $r = 3, 4, \ldots, K$, can be extended into one that is feasible also for $\mathcal{C}_r$ by fixing the schedule for nodes in $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{r-1}$ and then running an auction algorithm for the nodes in $\mathcal{C}_r$ with modified utilities

$$V^{(r)}(w, c_r) = \sup_{c_{r+1}, \ldots, c_K} \{U(\{\mathcal{S}_{\mathrm{w}}(1), \ldots, \mathcal{S}_{\mathrm{w}}(r-1), c_r, \ldots, c_K\}) - \mu_{c_{r+1}} - \cdots - \mu_{c_K}\}. \tag{13}$$

After enforcing primal constraints on nodes in all clusters up to and including $\mathcal{C}_K$, a feasible schedule has been generated from the solution to (11), and we can compute its primal objective function value using (8).

4.2.3. Algorithm Termination Criteria. By the weak duality theorem [19, Chapter 6], we have that, for any feasible schedule $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_W\}$, and any $\boldsymbol{\mu}$,

$$q(\boldsymbol{\mu}) \geq \sum_{w=1}^{W} U(\mathcal{S}_w). \tag{14}$$

We denote the optimal primal objective function value by $f^\star$, and the objective function value computed at iteration ν by $f^{(\nu)}$. Then, at iteration i,

$$\min_{\nu \in \{0, \ldots, i\}} q\left(\boldsymbol{\mu}^{(\nu)}\right) - \max_{\nu \in \{0, \ldots, i\}} f^{(\nu)} \geq f^\star - \max_{\nu \in \{0, \ldots, i\}} f^{(\nu)}. \tag{15}$$

The difference $\min_{\nu \in \{0, \ldots, i\}} q(\boldsymbol{\mu}^{(\nu)}) - \max_{\nu \in \{0, \ldots, i\}} f^{(\nu)}$ is called the duality gap of iteration i, and it upper bounds the distance from the so far best found primal objective function value to the supremum of the primal objective function. A measure of the quality of the best found solution up to iteration i is the relative duality gap, given by

$$\Delta(i) = \frac{\min_{\nu \in \{0, \ldots, i\}} q\left(\boldsymbol{\mu}^{(\nu)}\right) - \max_{\nu \in \{0, \ldots, i\}} f^{(\nu)}}{\max_{\nu \in \{0, \ldots, i\}} f^{(\nu)}}. \tag{16}$$

If we find a primal feasible schedule with a zero, or small, duality gap, then this schedule is guaranteed to be optimal, or near-optimal, in (8). On the other hand, if the best found schedule has a large relative duality gap, for example, $\Delta = 1$, then we know that a significant increase in the objective function value could be possible if we continue iterations. The proposed algorithm terminates when $\Delta(i)$ falls below a threshold (or when a maximum number of iterations have been exceeded).

4.2.4. Dual Variable Update Step. If a satisfactory schedule has not yet been obtained, we update dual variables in $\boldsymbol{\mu}$. We use the heuristic "price-update" method proposed in [18], which is loosely based on the subgradient method [19] and has been shown to perform well for similar problems. We only give a brief overview of the update method here, and refer to [18] for the details.

After step (1) in iteration i, a node $c_k \in \mathcal{C}_k$ will be temporarily "scheduled" in g_{c_k} slots. Clearly, the scheduling constraint is only satisfied if and only if $g_{c_k} = 1$. In step (2), the constraints will be enforced, cluster by cluster, by successive auctions. Let p_{c_k} be the price of the node c_k after the auction. We form three vectors, $\boldsymbol{\mu}_k^{(i)} \in \mathbb{R}^W$, $\mathbf{g}_k^{(i)} \in \mathbb{Z}^W$, and $\mathbf{p}_k^{(i)} \in \mathbb{R}^W$, whose elements are $\mu_{c_k}^{(i)}$ (the dual variable for node c_k at iteration i), $g_{c_k}^{(i)}$, and $p_{c_k}^{(i)}$, respectively. The update rule for the dual variables at iteration i is then

$$\boldsymbol{\mu}_k^{(i+1)} = \boldsymbol{\mu}_k^{(i)} - \left| \min_{\nu \in \{0,\ldots,i\}} q\left(\boldsymbol{\mu}^{(\nu)}\right) - \widetilde{q}_k\left(\boldsymbol{\mu}^{(i)}\right) \right| \frac{\mathbf{p}_k^{(i)}}{\overline{\mathbf{p}}_k^{(i)}} \odot \frac{\mathbf{g}_k^{(i)}}{\left\|\mathbf{g}_k^{(i)}\right\|^2}, \tag{17}$$

where $k = 3, 4, \ldots, K$, $\overline{\mathbf{p}}$ is the average of elements in $\mathbf{p}$, $\|\cdot\|$ denotes Euclidean norm, $\odot$ denotes element-wise multiplication, and

$$\widetilde{q}_k\left(\boldsymbol{\mu}^{(i)}\right) = \max_{\{\mathcal{S}_1(k),\ldots,\mathcal{S}_W(k)\}} \sum_{w=1}^{W} V^{(k)}(w, \mathcal{S}_w(k)) + \sum_{c_{k+1} \in \mathcal{C}_{k+1}} \mu_{c_{k+1}} \cdots + \sum_{c_K \in \mathcal{C}_K} \mu_{c_K} \tag{18}$$

such that

$$(\mathrm{B}_k) \quad \sum_{w=1}^{W} \mathbf{I}_{\mathcal{S}_w(k), c_k} = 1, \quad \forall c_k \in \mathcal{C}_k.$$

Note that $\widetilde{q}_k(\boldsymbol{\mu}^{(i)})$ is implicitly obtained when using the auction algorithm to enforce constraints on nodes in $\mathcal{C}_k$.

Intuitively, this dual variable update approach can be interpreted as follows. If, after fixing the schedule for clusters $\mathcal{C}_1$ to $\mathcal{C}_{k-1}$, two or more slots have a given node in $\mathcal{C}_k$ as their preferred choice in terms of interference conditions, then the "price" of this node is increased in future iterations of the algorithm. If there exist a node that no slot has as its preferred choice, then the price of this node is reduced. This way, solutions to the relaxed problem (11) that violates constraints in (8) are penalized.

For further examples of this dual method, although in a different application, we refer to [18] and references cited therein. A flowchart of the proposed algorithm is shown in Figure 2.

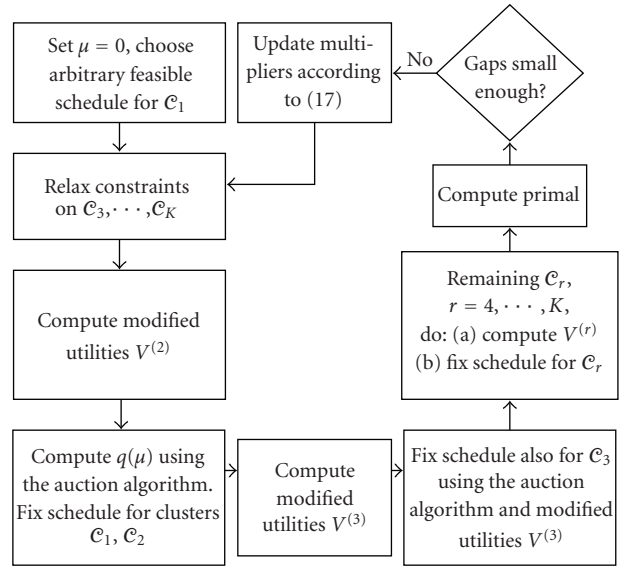

Figure 2: Flowchart of the proposed algorithm.

5. Numerical Analysis and Discussion

5.1. System Setup. We consider a short-range clustered WSN, where all transceiver nodes use BPSK signalling with a fixed output power P. 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:

$$\overline{P}_{i,k} = P_0 \left(\frac{d_0}{d_{i,k}}\right)^{\alpha}, \tag{19}$$

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, $P_0/P_N = 10\,\mathrm{dB}$ at reference distance $d_0 = 1\,\mathrm{m}$ and the path-loss exponent is $\alpha = 4$. All links are affected by Rayleigh fading with unit power gain, that is assumed to be independent between links. We assume that thermal receiver noise and interference are both Gaussian with zero mean. A simple $t = 5$ bit error correcting block code with block length $L = 800$ bits is used by all nodes. One codeword is transmitted in each time slot, it occupies the entire slot, and the cluster head uses hard-decision decoding of codewords. All nodes are assigned one slot per frame, and this slot assignment does not change between frames in the simulation. We assume that the frame of cluster i starts at global time $T + \omega_i$, where the synchronization errors $\{\omega_i\}_{i=1}^{K}$ are i.i.d. zero-mean Gaussian with standard deviation σ_ω. An example network schedule with $W = 3$ slots and $|\mathcal{C}_k| = 3$ nodes per cluster ($k = 1, 2, 3$) is shown in Figure 3, where $s_{k,w}$ denotes the node in $\mathcal{C}_k$ scheduled in slot w, L is the number of symbols per packet, and T_s is the symbol duration. We remark that robustness to cluster synchronization errors can of course be increased if guard intervals are introduced between slots in the frame structure.

To emulate a network configuration where a clustering algorithm, for example, LEACH [2], has been executed,

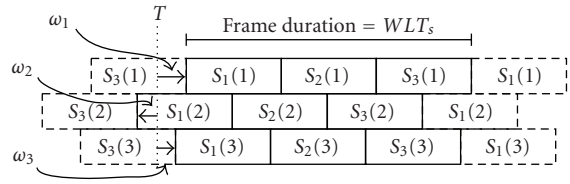

Figure 3: Example of cluster frames and synchronization errors.

we first manually deploy K cluster heads at coordinates $\{(x_i, y_i)\}_{i=1}^{K}$. The node coordinates in the kth cluster is drawn as $|\mathcal{C}_k|$ realizations from a circular Gaussian distribution with mean equal to the coordinates of the kth cluster head and standard deviation σ_R. The distance between cluster heads, σ_R, α, and P_0/P_N together determines the expected SINR conditions in the network. For a fixed α and P_0/P_N, a network with sparsely deployed cluster heads and small σ_R on the average experiences less intercluster interference than a network with more dense cluster heads and/or higher σ_R.

For notational convenience, we define a constant transmission range R, which is the range where the packet error rate (PER) goes above 10^{-2} in absence of fading and interference. In all simulated scenarios discussed below, cluster heads were deployed on the corners of a square with side R, that is, at $(0,0)$, $(R,0)$, $(0,R)$, and (R,R). Each of the $K = 4$ clusters has 5 nodes, and there are $W = 6$ slots in a frame, which implies that each cluster has one dummy node. The proposed scheduling algorithm was run until the relative duality gap $\Delta(i) \leq 10^{-3}$, see (16), or until a maximum of 300 iterations.

5.2. Convergence Properties of the Proposed Algorithm and Some Remarks on Complexity. The convergence of Lagrangian relaxation method that is used in this work is only guaranteed if a strong duality property can be proven. Since strong duality of the general method used here is still an open problem in the literature, the convergence of this application of the optimization method is not proven either. Nevertheless, the dual function, defined in (11), provides an upper bound to the primal which implies that even if the algorithm fails to converge in some scenarios, the maximum potential gain of an unknown optimal solution over the best known schedule is always obtained. This result has significant practical importance as it can be used to trade performance for complexity, especially when working with iterative scheduling approaches.

In this section, the convergence properties of the proposed algorithm are studied by extensive simulations. During the simulations, the best achieved utilities $U^{(1)}$, $U^{(5)}$, and $U^{(300)}$ after at most 1, 5, and 300 iterations, respectively, were stored. An upper bound on achievable utility in each network was also computed. This bound was computed as $(1 + \Delta(300))U^{(300)}$, where Δ is the relative duality gap defined in (16). To investigate the convergence properties of the proposed algorithm, we computed the relative utility difference $\epsilon(i) = (1/M)((1 + \Delta(300))U^{(300)} - U^{(i)})/U^{(i)}$, for $i = 1, 5, 300$. This difference indicates how close to the optimal solution the algorithm is after i iterations. If $\epsilon = 0$, then the optimal schedule has been found, while, if $\epsilon > 0$, an average relative increase of ϵ in utility per node could possibly be achieved by additional iterations of the algorithm. We define $\epsilon(0)$ to be the relative difference between a random scheduling (with utility $U^{(0)}$) and the upper bound.

The CDFs of $\epsilon(i)$, $i = 0, 1, 5, 300$, for cluster densities $\sigma_R = R/2$ and $\sigma_R = R/4$, are plotted in Figure 4. We note that the achieved objective function value is very close to its upper bound after at most the maximum 300 iterations. In fact, in our simulations, the $\epsilon(300)$ was almost always less than 0.001. Hence, the algorithm performs well in terms of convergence. Additionally, Figure 4 indicates that convergence is quite fast, that is, the distance to the upper bound is reasonably small already after only a few iterations.

The results obtained after a single iteration deserves some special attention. It appears that a reduced complexity "greedy" algorithm that only iterates once, that is, executes $K - 1$ consecutive auction algorithms, can be used without a significant degradation in performance. This conclusion is of great importance in networks where complexity is a limiting factor. The overall complexity of the proposed algorithm mainly depends on W, K, and the number of iterations the algorithm spends before termination. As noted in Section 4.2, there are $(W!)^K$ different feasible schedules to consider. However, the proposed algorithm only investigates a small subset of all possible feasible schedules.

Empirical tests on a personal computer have indicated that networks of up to $K = 10$ clusters, each with $W = 7$ nodes (e.g., the maximum number of slaves in a Bluetooth network), are manageable with the proposed algorithm. The part of the algorithm that introduces most complexity is the search for $V^{(2)}(w, c_2)$ in (11), which is implemented here as an exhaustive search over the W^{K-2} possible relaxed slot w assignments. If larger networks than $K = 10$, $W = 7$ is required, then more advanced search methods must be considered.

Since all the nodes in every clusters are scheduled after a single run of the algorithm, the update frequency of the schedules depends only on the mobility, that is, the rate that average powers vary. In the low mobility sensor networks considered in this paper, the frequency of schedule update is substantially lower than the schedule usage time and therefore, the communication overhead cost of the proposed algorithm in these scenarios is negligible.

5.3. Throughput in a Perfectly Synchronized Scenario. If the network layout is such that the intercluster interference is low, or the distance between node and sink is too long for communication even in the interference-free case, the benefits of using the proposed algorithm compared to just using an arbitrary schedule should intuitively be quite small (as an extreme case, consider a network where σ_R approaches zero, or goes to infinity). To quantify this, 200 networks with $\omega_i = 0$, $i = 1, \ldots, K$, and $R = 1.25$ m were generated. The

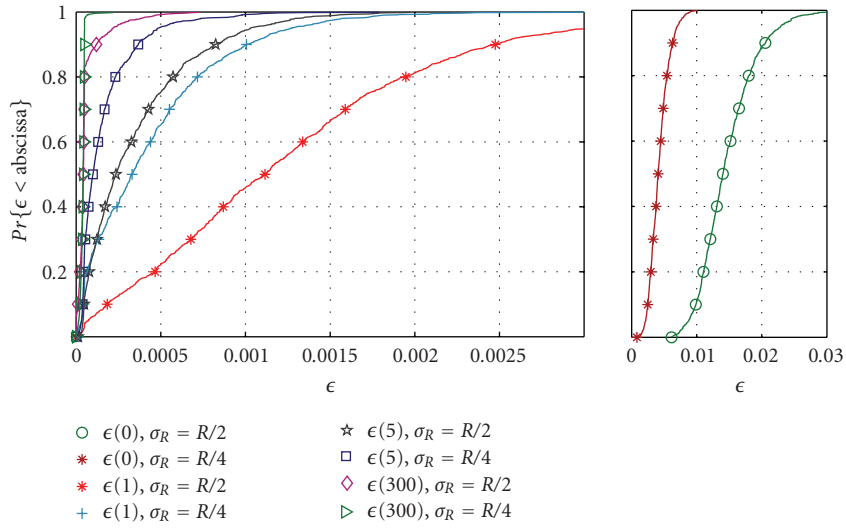

Figure 4: Algorithm convergence, $\epsilon(i) = (1/M)(((1 + \Delta(300))U^{(300)} - U^{(i)})/U^{(i)})$.

threshold was again set to $\Theta = 4.82$ dB. Two schedules were generated for each network layout, one using the proposed algorithm, and one random but feasible schedule. For each network and schedule, in addition to $\overline{P}_{e,m}$ for all nodes, we also compute the normalized network throughput T, given by $T = M^{-1} \sum_{m=1}^{M} (1 - \overline{P}_{e,m})$.

The CDF of network throughput is plotted in Figure 5 for the two scheduling approaches and for three different cluster densities. As expected, the increase in throughput is the highest when interference is significant, but SINR is still sufficient for communication. The increase in throughput is slim for scenarios where the overall SINR conditions in the network are either very good, or very bad, compare with the results for $\sigma_R = R/8$ and $\sigma_R = 2R$.

To investigate the impact of the proposed scheduling approach on individual nodes, we also plot the CDF of $\overline{P}_{e,m}$ evaluated for all nodes in all networks (Figure 6). It is interesting to note that the proposed algorithm reduces the number of nodes with relatively high $\overline{P}_{e,m}$, at the expense of nodes that have a $\overline{P}_{e,m}$ closer to zero. Although this effect is not significant, it is noticeable for the case of $\sigma_R = R/2$.

As mentioned in Section 3, the utility measure in (7) does not need to encompass all clusters. For instance, suppose we control a number of node clusters that are deployed in the vicinity of a number of "alien" clusters with a fixed TDMA schedule that we have knowledge of, but cannot control. We would like to maximize the packet delivery ratio in our network, but we may not want to do so at the expense of the "alien" clusters, that could for instance be a legacy system. The impact on PER in clusters that are not accounted for by the utility function was evaluated in 200 networks. The parameters used in Figures 5 and 6, with $\sigma_R = R/4$, were also used here. We assume that we can control the schedule of clusters with cluster heads at $(0, R)$ and (R, R), while cluster heads at $(0, 0)$ and $(R, 0)$ choose a random feasible schedule for the nodes in their corresponding clusters. In Figure 7, the CDFs of $\overline{P}_{e,m}$ in the controlled and uncontrolled

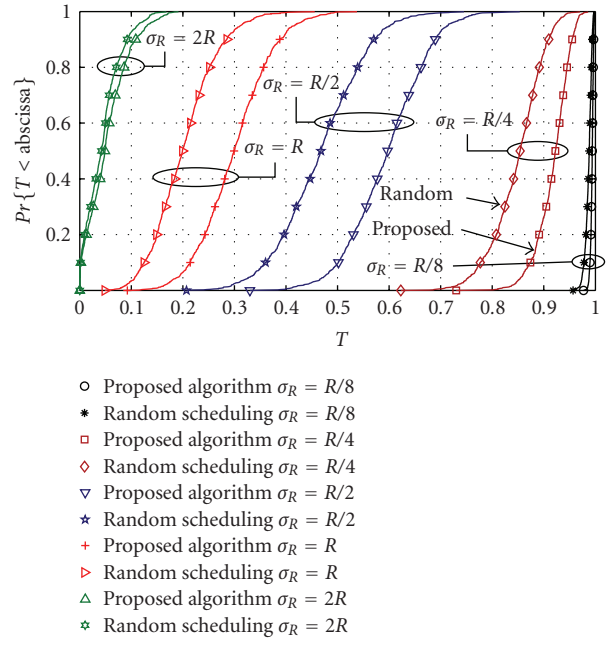

Figure 5: CDF of network throughput T for random and proposed scheduling.

clusters are compared. The CDF of $\overline{P}_{e,m}$ in a network where all cluster heads arbitrarily choose their schedule is also shown. Somewhat surprisingly, we see that $\overline{P}_{e,m}$ in uncontrolled clusters does not differ significantly from what would be experienced if all four cluster heads would have arbitrarily chosen feasible schedules. Hence, throughput in the uncontrolled clusters is not significantly degraded by the "smart" scheduling made in controlled clusters.

5.4. Packet-Error Rates with Cluster Synchronization Errors. Up to now, we have assumed that clusters are perfectly

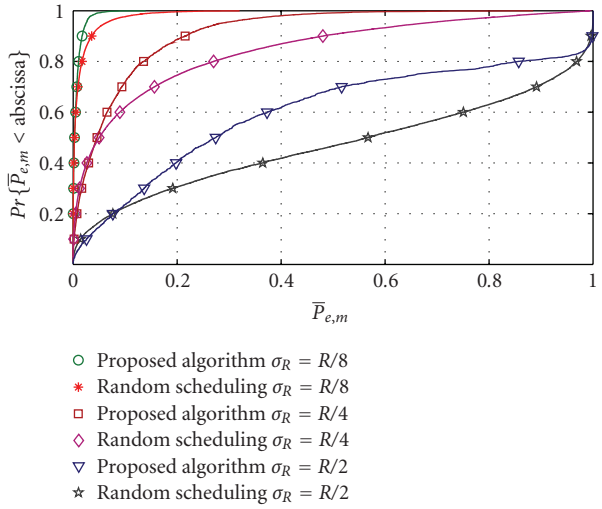

FIGURE 6: CDF of $\overline{P}_{e,m}$ for all nodes in all networks.

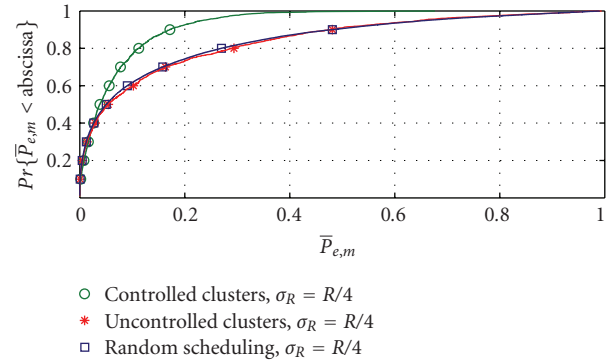

FIGURE 7: CDF of $\overline{P}_{e,m}$ when scheduling a subset of clusters.

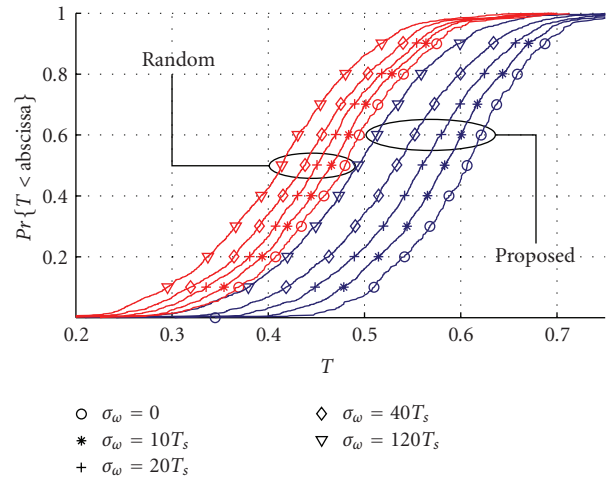

FIGURE 8: CDF of network throughput T, with synchronization errors.

synchronized in time, so that all slots begin and end simultaneously. We have also neglected the propagation delay between nodes in the network when computing the instantaneous SINR. Obviously, these assumptions will not hold in a real network. We therefore investigated the impact of cluster synchronization errors on throughput in 400 networks with the same setup as in Section 5.3. To account for synchronization errors and varying propagation delays, zero-mean Gaussian synchronization errors $\{\omega_i\}_{i=1}^{K}$ with standard deviation σ_ω were introduced (see Figure 3). The CDF of network throughput is plotted in Figure 8 for $\sigma_R = R/2$. As expected, synchronization errors reduce network throughput when using the proposed algorithm, but also when using a random schedule. Note that, even for relatively large errors, for example, $\sigma_\omega = 40T_s$ (corresponding to a relative error standard deviation between two clusters of $80T_s$, which is 10 percent of the packet duration), the degradation is not overly severe, and we see a significant gain in throughput over a random scheduling. Simulation results not shown here also indicated that the robustness to synchronization errors is higher in networks with better SINR conditions, for example, for $\sigma_R = R/4$.

6. Conclusions

We have, by modelling interference as additive and Gaussian, derived an expression for the packet-loss probability in networks with mutually interfering clusters of transceivers deployed in Rayleigh-fading environments. Computer simulations showed a good agreement between the model and actual packet error-rates.

A scheduling algorithm for clustered wireless networks that exploits the derived packet-loss model was then presented. Computer simulations of networks with transmissions scheduled by the proposed algorithm showed that a significant increase in network throughput is achievable as compared to the case where clusters choose schedules independently without considering the schedules at interfering clusters. Although the scheduling algorithm was derived under the assumption of a perfectly synchronized network, we have shown that a synchronization error on the order of several symbol durations does not degrade the algorithm performance significantly.

Numerical results indicate that convergence to the optimal schedule almost always occur with a reasonable number of iterations. Hence, the proposed algorithm can be used as a tool for benchmarking the performance of other (suboptimal) scheduling algorithms.

Appendix

Packet-Loss Probability

Without loss of generality, let the node $\mathcal{S}_w(1)$ transmit to the sink in $\mathcal{C}_1$. For notational convenience, let $\gamma_0 = \Theta(P_{N_1}/\overline{P}_{\mathcal{S}_w(1),1})$, $\gamma_j = \Theta(\overline{P}_{\mathcal{S}_w(j),1}/\overline{P}_{\mathcal{S}_w(1),1})$, and $A = \gamma_0 + \sum_{j=2}^{K} \gamma_j a_j$, where a_j denotes the fading coefficient from the

node $\mathcal{S}_w(j)$ to the sink in cluster 1, that is, $\kappa_{S_w(j),1}$. Starting from (2), we have

$$\begin{aligned}
&P_{\text{loss}}(k, \mathcal{S}_w) \\
&= \Pr\{\Gamma(k, \mathcal{S}_w) < \Theta\} = \Pr\{a_1 < A\} \\
&= \int_{a_K=0}^{\infty} \cdots \int_{a_2=0}^{\infty} \int_{a_1=0}^{A} \exp\left(-\sum_{i=1}^{K} a_i\right) \mathrm{d}a_1 \cdots \mathrm{d}a_K \\
&= \int_{a_K=0}^{\infty} \cdots \int_{a_2=0}^{\infty} \exp\left(-\sum_{i=2}^{K} a_i\right) (1 - \exp(-A)) \mathrm{d}a_2 \cdots \mathrm{d}a_K \\
&= 1 - e^{-\gamma_0} \int_{a_K=0}^{\infty} e^{-(1+\gamma_K)a_K} \cdots \int_{a_2=0}^{\infty} e^{-(1+\gamma_2)a_2} \mathrm{d}a_2 \cdots \mathrm{d}a_K \\
&= 1 - \frac{e^{-\gamma_0}}{\prod_{j=2}^{K} \left(1 + \gamma_j\right)},
\end{aligned} \tag{A.1}$$

where we have used the assumption that the fading coefficients are i.i.d. with unit mean. The last equality follows from the identity $\int_0^{\infty} e^{-ax} \mathrm{d}x = a^{-1}$, that holds for $a > 0$.

Acknowledgment

This work was supported by Vinnova Project no. 2003-02803.

References

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons, New York, NY, USA, 2006.

[2] 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, pp. 660–670, 2002.

[3] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

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

[5] J. Li and G. Y. Lazarou, "A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 55–60, 2004.

[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] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in wireless sensor networks: distributed edge-coloring revisited," *Journal of Parallel and Distributed Computing*, vol. 68, no. 8, pp. 1122–1134, 2008.

[9] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for dynamic multi-server wireless systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1506–1515, 2006.

[10] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Joint opportunistic power scheduling and end-to-end rate control for wireless ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 801–809, 2007.

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

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

[13] M. R. D. Rodrigues, I. Chatzigeorgiou, I. J. Wassell, and R. Carrasco, "Performance analysis of turbo codes in quasi-static fading channels," *IET Communications*, vol. 2, no. 3, pp. 449–461, 2008.

[14] S. Lin and D. J. Costello, *Error Control Coding*, Pearson Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1982.

[15] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM, Philadelphia, Pa, USA, 1983.

[16] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

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

[18] S. Deb, M. Yeddanapudi, K. Pattipatti, and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensormultitarget state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 523–538, 1997.

[19] N. Andréasson, A. Evgrafov, and M. Patriksson, *An Introduction to Continuous Optimization*, Studentlitteratur, 2005.