Self-Stabilizing MAC Protocols for Large-Scale, Heavy Loaded Sensor Networks under Consideration of Hidden Nodes

Stefan Unterschütz and Volker Turau

Hamburg University of Technology
stefan.unterschuetz@tu-harburg.de, turau@tu-harburg.de

Abstract. In large-scale, heavy loaded sensor networks the hidden node problem significantly restricts the attainable throughput. This paper examines this issue and depicts why most TDMA as well as dedicated hybrid MAC protocols are still negatively affected by this phenomenon. The concept of probabilistic self-stabilization is adopted to provide a framework for implementable reservation MAC protocols that avoid packet loss caused by signal interferences even under high load. These protocols base upon two main primitives: continuity in channel access, allowing predictability, and acknowledgments, permitting to discover packet loss. The designed TDMA and CSMA protocols are able to cope with the hidden node problem and with topology changes and achieve a high throughput in a steady state. The protocols are simulated and compared with IEEE's 802.15.4 unslotted CSMA/CA protocol.

1 Introduction

Wireless sensor networks are an appropriate technology for a large-scale, non-invasive observation of physical or ecological state variables like temperature, air pressure or movement. A sensor network consists of a large number of tiny wireless sensor nodes transferring measured data to predefined base stations. Current research is focused on energy-efficient communication techniques suited for battery powered devices. Recent progress in energy-harvesting or wake-up receivers lessens the demand for energy-efficient solutions. In such scenarios requirements like minimum delay for event-reporting and high throughput move back into the center of attention. This motivates the development of novel communication protocols for large-scale sensor networks.

Since sensor nodes share the same radio channel, the media access control (MAC) protocol plays a key role for attaining a high throughput, low latency, and fairness. A concurrent channel access that leads to packet loss caused by signal interference at a receiver side has to be avoided. For solving this task, two main paradigms have been proposed: CSMA (carrier sense multiple access) and TDMA (time division multiple access).

In TDMA protocols nodes are synchronized and access the channel only in predefined time slots. Here, the challenge is to find a conflict-free slot assignment

by additionally providing a high spatial reuse of slots. Unfortunately, finding a valid schedule for large-scale networks is often associated with a high controlling overhead. Furthermore, existing localized approaches based on coloring (e.g. of the 2-hop neighborhood) are inappropriate in real world scenarios, because assumptions such as the unit disk model or bidirectional links do not hold. Further drawbacks of most TDMA protocols are the scarce ability to cope with topology changes and the high latency.

Another paradigm for channel access is CSMA/CA, which is popular due to its simplicity and flexibility. A sending trial starts with a random delay (offset) and, in contrast to the ALOHA protocol, a subsequent sensing of the channel state. In case of a none occupied channel data is sent, otherwise transmission is deferred. This scheme doesn't solve the problem of hidden nodes, hence a signal interference of two sending nodes at a receiver may occur, although the sending nodes are not in each other communication or sensing range. The announcement and confirmation of a scheduled data transmission (RTS/CS) can reduce the effect of hidden nodes, but is inefficient if only small-sized packets have to be transmitted. In general the hidden node problem becomes the bottleneck for CSMA schemes in multi-hop networks with a high density and high data rates.

Neither common TDMA nor CSMA approaches are suitable for dense sensor networks with a high traffic demand. For such scenarios we propose two self-stabilizing MAC protocols. These protocols are fault-tolerant and aware of traffic flow. They automatically allocate the required bandwidth which is particularly beneficially in data gathering applications.

For the MAC protocol we borrow concepts of reservation ALOHA, TDMA, and the theory of probabilistic self-stabilizing. Like in pure or slotted ALOHA the channel is randomly accessed. Possible packet collisions are detected by using acknowledgments. Additionally, time is organized in super-frames in which a communication attempt is done with a random offset. If the transmission succeeds, a node tries to reuse this offset for further communication trials. In case of communication faults (e.g. caused by hidden nodes) the sender randomly picks a new offset. The theory of probabilistic self-stabilizing guarantees that eventually a steady state is reached. Different enhancements of the protocols are possible: Based on additional information acquirement through idle listening nodes can decrease stabilization time and the assignment of multiple slots to a node increases throughput and fairness. Our algorithms are analyzed and simulated with respect to convergence, latency, maximum throughput.

This paper is structured as follows. In Sect. 2 current MAC-protocols for sensor networks and relevant research are reviewed. Then a suitable network model as well as a short analyses of the hidden node problem are presented. Based on this the inadequateness of current TDMA algorithms is shown and the idea of a probabilistic self-stabilizing MAC protocol is introduced. Afterwards we present in Sect. 4 the design of a TDMA and a CSMA protocol by adopting principles of the previous section. Subsequently the protocols are evaluated by simulations before we finally draw a conclusion.

2 Related Work

Recent MAC protocols for wireless sensor networks are optimized for low-power operations. Prominent examples are B-MAC [7] and X-MAC [3]. By introducing duty cycles, nodes can sleep and safe energy. The main drawback is the increased latency and low throughput.

A high throughput for wireless devices is provided by the IEEE 802.11 standard. In order to avoid collisions a four way handshake is applied. First a node sends a *ready to send* message (RTS) to the receiver, which replies with a *clear to send* (CTS), if no interferences are observed. Then all nodes in the range of the receiver and sender get to know that a data transmission is about to take place and will not disturb this. After the actual data is transmitted, the receiver replies with a final acknowledgment. Because of the controlling overhead caused by the RTS and CTS packages, this scheme is only profitable if the size of the data packets is large. Furthermore, hardware costs and high energy consumption makes this protocol unsuitable for sensor networks.

IEEE's 802.15.4 [1] is the common standard for wireless personal area networks (WPANs). Recent transceivers for sensor nodes are compliant to this protocol. By default 802.15.4 uses a CSMA approach (unslotted and slotted CSMA/CA), however, the protocol is highly configurable and can serve as the base for other MAC protocols. Nevertheless, the slotted as well as the unslotted CSMA/CA approach of 802.15.4 is vulnerable to the hidden node problem, specially in high density networks. Furthermore the slotted protocol is difficult to be applied and maintained in multi-hop networks due to the need of collision free beacon transmission [11].

Z-MAC [8] combines the strength of CSMA and TDMA. It achieves a high channel utilization under high contention exploiting a TDMA like approach. For low contention it behaves like CSMA protocols. The base of Z-MAC are broad time slots, which are used exclusively in case of high traffic and otherwise shared and accessed in a CSMA like fashion. The slot assignment is done at the time of deployment and thus the algorithm is vulnerable to topology changes. Examples for TDMA protocols that are intended for high-loaded sensor networks are Funneling MAC [2] and TreeMac [10]. They are traffic aware, but restricted to many-to-one communication, e.g. data gathering. In general TDMA approaches can efficiently avoid collisions. Main drawbacks are the need for synchronization and the challenge of finding a valid slot assignment. Z-MAC as well as TreeMAC assume bidirectional links and a communication range equal to the interference range.

For large-scale networks of high density the mentioned TDMA approaches are infeasible to realize. The theoretical basis of them are unit-disk graphs and multi-hop coloring. In fact, the electromagnetic radiation of antennas is highly heterogeneous and in interference-prone environments like urban areas the link states are unpredictable and altering. A proper modeling of the network is proposed by Ergen and Varaiya [5]. They distinguish between data flows and data interferences in order to construct a conflict graph which can be the base for different centralized coloring approaches. Although their network model is re-

stricted to many-to-one communication the idea of coupling traffic flow and physical interferences is adopted and generalized in this paper.

Finally, the principles of probabilistic self-stabilization [6] are used to design a distributed MAC protocol. Self-stabilization ensures that starting from an arbitrary system state, a legitimate state is eventually reached in a finite number of steps. For probabilistic algorithms there is no upper bound for steps required to terminate, but expectation values can be derived. The advantage of probabilistic self-stabilizing algorithms is that they often have a simple construction and low memory demand by still providing fault-tolerance.

3 Analysis

In this section we address the issue of the lack of dedicated network and communication models for describing signal interferences caused by other sensor nodes. In this context we prove the non-existence of scalable, conflict-free, and deterministic slot assignment protocols and the need of novel solutions to allow a conflict-free access of the wireless channel.

3.1 Network and Communication Model

In the literature a sensor network is often modeled as a graph $G = (V, E)$. V represents the set of wireless nodes, where the network size is given by $N = |V|$. Furthermore, E represents a set of bidirectional links between nodes. Protocols developed for this model may not work for real wireless networks, in which unidirectional links as well as interference beyond the communication radius are observable. If these phenomena are ignored no proper handling of the hidden node problem can be achieved.

To be able to make an adequate analyses of the hidden node problem and the achievable network performance, we describe a wireless network by an interference graph $G_{\text{if}} = (V, E_{\text{if}})$ and a data flow graph $G_{\text{flow}} = (V, E_{\text{flow}})$. The interference graph G_{if} indicates whether the transmission of data by node v_1 disturbs the reception of data at node v_2. In this case $(v_1, v_2) \in E_{\text{if}}$. A similar model for trees can be found in [5]. In general, E_{if} is a superset of the set of bidirectional links E which implies that E_{if} additionally contains edges for which a reliable transmission is impossible. Furthermore, a directed edge $(v_1, v_2) \in E_{\text{flow}}$ of the data flow graph G_{flow} depicts whether a direct exchange of data from node v_1 to node v_2 is required. G_{flow} depends on the routing protocol, however if nothing is known about the traffic flow, G_{flow} can be set equal to E, meaning that data may be transmitted by a node to all neighbors. In this paper reliable data transmission is considered between two nodes on behalf of the MAC, thus if v_1 sends data to v_2, then node v_2 must be able to send an acknowledge to v_1. Thus, G_{flow} is symmetric. The main benefit of considering the flow of data is the possibility to determine the maximum possible spatial reuse of the radio channel.

Based on G_{flow} and G_{if} a conflict graph $G_{\text{cfl}} = (V, E_{\text{cfl}})$ can be calculated. The conflict graph represents connections between nodes, for which a simultaneous data transmission can produce a collision on the receiver side. G_{cfl} contains hidden nodes as well as possible communications conflicts that could be resolved by using carrier sense techniques. The set of edges of the conflict graph is defined by the union of $E_{\text{cfl}} = E^1_{\text{cfl}} \bigcup E^2_{\text{cfl}}$.

E^1_{cfl} contains an edge between a node v_1 and v_3, if the transmission of data from v_1 to v_2 can be disturbed by v_3. Note that in case of transmission with acknowledgements E^1_{cfl} becomes symmetric.

$$E^1_{\text{cfl}} = \{\, (v_1, v_3) | v_1, v_3 \in V \wedge \exists v_2 \in V \,((v_1, v_2) \in E_{\text{flow}} \wedge (v_3, v_2) \in E_{\text{if}})\,\} \quad (1)$$

We are assuming half-duplex connections, thus a concurrent transmission of data of two nodes v_1 and v_2 is leading to a conflict:

$$E^2_{\text{cfl}} = \{\, (v_1, v_2) | (v_1, v_2) \in E_{\text{flow}}\,\} \quad (2)$$

Additionally let $N^{\text{cfl}}_v = \{n \in V | (v, n) \in E_{\text{cfl}}\}$ be the set of neighboring nodes of v being in conflict with v.

Figure 1 depicts an example for the presented notation. Note that node v_3 and v_6 are in interference range, but not in conflict, meaning that a concurrent transmission of both do not lead to a packet collision.

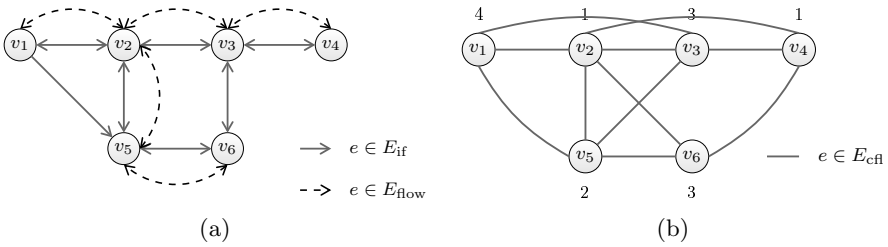


Fig. 1. Example of an interference and flow graph as well as the resulting conflict graph

3.2 Collision-Free Media Access

Based on G_{cfl} the condition for a collision-free media access can be derived. Free of collisions in this context means that no packet loss occurs due to signal interferences of two sending nodes.

Theorem 1. *A media access scheme, where no neighboring nodes in E_{cfl} send concurrently, is completely free of collisions.*

Proof. (By Contradiction) Suppose there is a node pair (v_1, v_2) not in E_{cfl} for which a collision occurs if both nodes are sending concurrently. Without loss of generality v_1 transmits data either to v_2 or to another node v_3. The former case fulfills (2) and contradicts with $(v_1, v_2) \notin E_{\text{cfl}}$, thus v_1 must send data to a node v_3. A packet collision caused by v_2 at v_3 can only occur if and only if $(v_2, v_3) \in E_{\text{if}}$. This case was considered in (1). Thus, (v_1, v_2) has to be in E_{cfl}. □

As a result of Theorem 1, a collision free slot assignment for TDMA protocols is a valid coloring of G_{cfl}. The lower bound for the number of required colors is the chromatic number. Identifying the chromatic number for a graph is NP-complete. For this reason a heuristic and distributed slot assignment should be preferred in large-scale networks. A wide variety of algorithms accomplish slot assignment by coloring the two or even three hop neighborhood of G [8, 10] in lieu of G_{cfl}. Unfortunately finding a coloring for G is missing the target of finding collision free assignments in real wireless networks. Also the achievable throughput may not be optimal, because neighbours in E are not necessarily neighbours in E_{cfl}, thus a a slot reuse is possible.

It can be shown that an efficient non-centralized TDMA protocol for G_{cfl} can not exist.

Theorem 2. *Finding a valid coloring with less than N colors for an arbitrary conflict graph G_{cfl} has a complexity of $\Omega(N)$ rounds for a distributed, deterministic algorithm.*

Proof. In order to exclude a trivial slot assignment algorithm by using unique identifiers, a valid coloring with less than N colors is assumed, if existent. Consider an open ring topology which is disconnected by two nodes that are assumed to be in interference range, but not in communication range. For these nodes checking if an assigned slot is valid takes at least N rounds, because the information has to flow from one end of the open ring to the other. □

From this it follows that no efficient and deterministic slot assignment for large-scale networks exists that provides a collision free channel access. Nevertheless finding a collision-free channel access scheme is the key challenge in order to reach a high throughput.

3.3 Probabilistic Self-Stabilizing Slot Assignment

For most applications a reliable transmission of data is used, meaning that the transmission is secured via an acknowledgement. This allows a node to experience the occurrence of collisions assuming a stable link (no packet loss due to multipath propagation). Beside this we exploit continuity in accessing the channel. In this context continuity allows neighboring nodes to be able to predict when a node transmits data. In fact the latter is the basic concept of all TDMA algorithms, too.

The primitives continuity and acknowledgments are combined using a probabilistic stabilizing algorithm. It is assumed that all nodes are synchronized and have a total of κ possible slots for communication. Each time slot has exactly the length which is necessary for sending equal length data packets with a subsequent reception of an acknowledgment. Note that the introduction of immediate acknowledgments must be considered in the construction of G_{cfl}, in which a sender v_1 originates the reply of an acknowledgment by a receiver v_2 in the same time slot. On start-up or in case of detected collisions a node randomly selects a new sending slot.

Theorem 3. *The proposed algorithm probabilistically stabilizes towards a collision free slot assignment if* $\kappa \geq \Lambda^{\text{cfl}} + 1$ *with* $\Lambda^{\text{cfl}} = max_{v \in V}(|N_v^{\text{cfl}}|)$.

Proof. First, a valid coloring of G_{cfl} exists, because even if all neighbors N_v^{cfl} of node v have different colors at least one color is available which is guaranteed by $\kappa \geq \Lambda^{\text{cfl}} + 1$. In the proposed algorithm a node randomly picks a new slot if it is in conflict. Here the lower bound for the probability to select a valid slot is $(\frac{\kappa - \Lambda^{\text{cfl}}}{\kappa})$. The joint probability for a valid slot assignment of all nodes is bounded below by $P_{val} \geq (\frac{\kappa - \Lambda^{\text{cfl}}}{\kappa})^N$. Now the probability for not reaching a valid state after m steps is $(1 - P_{val})^m$ which converges for rising m against zero, thus the proposed algorithm probabilistically stabilizes. □

Obviously increasing κ decreases the expected convergence time, whereas the maximum channel throughput drops down, because slots may stay unused. In networks the maximum degree Λ^{cfl} of the conflict graph is often unknown making a generous estimation for the upper bound necessary.

3.4 Calculation of the Markov Chain for the Slot Assignment

A calculation of the expectation value of the convergence time is done by using a time-homogeneous Markov chain. The structure of the transition matrix bases on the graph G_{cfl} and the proposed slot assignment algorithm of the previous section. Let $S = \{1, \ldots, \kappa\}^N$ be the state space of the Markov chain. The state space contains each combination of possible slot assignments for all nodes. Let $S_{leg} \subseteq S$ be the set of legal states for which no conflict occurs, so a valid coloring of G_{cfl}. The transition Matrix $M = (p_{ij})$ is $p_{ij} = P(X_{n+1} = S_j | X_n = S_i)$ for the random variables $X_1, X_2 \ldots$. For a legal state S_i the transition probabilities are $p_{ii} = 1$ and $p_{ij} = 0, i \neq j$. Hence, a legal state is a fixpoint assuming that no faults occur. Let S_i be a non legal state and let $S_i^* \subseteq S$ a set of possible following states. For example if S_i contains exactly one conflict between two nodes, then S_i^* contains all state combinations, which are equal to S_i except for the two conflict nodes, which can have an arbitrary newly assigned slot. For a random slot assignment $p_{ij} = \frac{1}{|S_i^*|}$, if the follow state $S_j \in S_i^*$, otherwise $p_{ij} = 0$. Let μ_0 the vector of the initial distribution of states, then the state distribution for the k-th step is $\mu_k = M^k \mu_0$.

Based on this result the expected value for the convergence time can be calculated. Let v be a vector $\{0,1\}^{|S|}$ containing a 1 at the i-th element if

$S_i \in S_{leg}$, otherwise zero. The probability to reach a legal state in the k-th step is:

$$P_{val}(Y = k) = p_k \prod_{i=0}^{k-1} (1 - p_i) \tag{3}$$

where p_n is defined as $p_n = vM^n\mu_0$, which is the probability of being in a legal state in the n-th step. Now the expected value of the convergence time can be calculated.

It should be noted that the definition of the needed transition matrix and the calculation of the expected convergence time is only feasible in small networks.

4 Design

In this section the principles of the previous section are adopted to design two self-stabilizing MAC protocols providing a high throughput in arbitrary networks.

4.1 Self-Stabilizing TDMA

The main concepts for designing a probabilistic TDMA protocol have been introduced in Sect. 3.3, details of the implementation in wireless sensor networks are treated in the following.

Time is separated in super-frames which are synchronized for all nodes. A super-frame itself is further partitioned into κ time slots. The length of a time slot is sufficient to send data and to receive afterwards an acknowledgment.

Each node randomly picks one slot for sending. This assignment stays unchanged until possible communication failures occur which are indicated by the absence of an acknowledgment. Here, basic concepts of the reservation ALOHA protocol [4] are adopted. A packet loss due to other faults like multi-path effects would trigger an unnecessary slot change. For this reason the execution of a reassignment can be implemented counter-based, meaning that an error has to occur multiple times, or probabilistic.

The convergence time towards a conflict free state is significantly enhanced by using idle listening. Currently if a node detects a collision it randomly picks a new slot. When selecting a slot that is already used by another node further collisions are due. Idle listening can be used to detect, whether a slot is free or already in use. For this purpose nodes listen to acknowledgments as an indication for used slots. By assuming bidirectional links, a sending trial in a slot where an acknowledgment is received may cause packet loss.

Since nodes are not using a time slot in each super-frame an recursive estimation of the slot state is applied. Not until a slots stays unused for several rounds it is marked as free. Note that the received data packets are not used for the estimation of the slot utilization, because (referring to Sect. 3.1) two adjacent nodes $(v1, v2) \in E_{\text{if}}$ are not consequently in conflict.

The bandwidth demand of nodes can be highly heterogeneous, e.g. in data gathering application a concentration of packets near the sink can be observed,

which is known as funneling effect [2]. For providing fairness we consider a multi-slot allocation approach, meaning that a node can be the owner of more than one slot. This has two main advantages: First, κ can be much greater than required (at the expense of the convergence time). But even more important is the fact that different nodes can utilize a different amount of bandwidth depending on their traffic demand (traffic awareness).

The allocation of new slots is done via the previously described probabilistic approach, but there is still the question how much slots a node is allowed to use. Based on empirical results we use the following flow-control technique: Let s_v be the number of valid time slots of node v, meaning that these slots are currently free of collisions. Let q_v be the number of packets in the queue of the node. A node allocates and deallocates a time slot if $q_v > s_v * 2$ and $q_v < s_v$ holds respectively. Obviously the implemented flow-control technique has significant influence on the performance of the protocol, however a detailed analyses of such algorithms is not part of this paper.

Multi-slot allocation allows to choose κ much greater than $\Lambda^{\text{cfl}}+1$. In general, the calculation of the required time slots κ is a challenging task for a network planner, since conflicts of nodes are difficult to identify a priori. If κ is too low, no stabilization is possible. On the other hand a high κ leads to reduced channel utilization. Applying multi-slot allocation allows to be generous in selecting κ. However, a high κ may lead to a higher delays due to the increased super-frame length.

A prerequisite for the self-stabilizing TDMA protocol is an underlying synchronization service to be able to align the slots among the nodes. Such services are well studied in research and several protocols for distributed algorithms are proposed [9]. Due to the synchronization overhead, the achievable data rate is reduced. In order to compensate synchronization errors, a guard interval is necessary for each time slot, which further reduces the bandwidth.

Broadcasts, for which commonly acknowledgements are suppressed, are not supported by the self-stabilizing TDMA protocol. Nevertheless, they can be realized by having dedicated broadcast slots or by randomly choosing a time slot.

4.2 Self-Stabilizing CSMA

The depicted self-stabilizing TDMA needs a global synchronization service. This causes additional overhead, which may be inadmissible in large-scale networks. Furthermore, the expected end-to-end delay is much higher than in a pure CSMA approach. The drawback of the self-stabilizing TDMA can be negotiated by composing the presented principles with a common CSMA protocol.

The self-stabilizing CSMA algorithm adopts the principles of the described TDMA algorithm of the previous section. Time is separated in super-frames, but these are not necessarily synchronized among the nodes. For simplicity all data packets are assumed to be of equal length. Each node is allowed to send multiple packets in its super-frame using a similar kind of flow-control technique which is explained for the self-stabilizing TDMA. Instead of using slots, a node tries to maintain fixed offsets for sending packets. If sending failed (i.e. no acknowledge is

received) or no data has to be send, the offset is abdicated. For the allocation of a new sending offset an ordinary exponential backoff algorithm is taken. A clear channel assessment can optionally be made at the beginning of each sending trial.

In order to reduce conflicts idle-listening is used to determine whether a possible sending period would lead to a collision. For this purpose received acknowledgments are used to mark the time offsets as occupied. Furthermore, an aging algorithm is used, so that after a given number of passed super-frames the time duration is marked as free, if no further data is received. Although an exponential backoff is used for the allocation of a new sending offset, further defering of the calculated backoff is done if possible conflicts with neighbors might occur. The self-stabilizing CSMA can also be used for broadcasting messages, whereby a random backoffing should be applied.

For low traffic-load the hop to hop delays are expected to be comparable with a normal CSMA algorithm. In this scenario nearly no congestion is observed and the list of fixed sending offsets is empty. If a packet has to be sent, an ordinary backoff algorithm selects a free sending offset. For high contention, nodes try to use conflict free backoffs or they revert to already allocated offsets, so the delay is increased. In case of high-loaded networks the behavior of the self-stabilizing CSMA approach is expected to be comparable with TDMA algorithms.

5 Simulation

In this section we examine and evaluate the proposed protocols. For this, simulations are performed to disclose the behavior of the protocols in respect to the metrics convergence, throughput, and delay. In this context also a comparison with the popular IEEE 802.15.4 unslotted CSMA/CA is also done. We selected the OMNeT++ framework [12] as simulation tool.

5.1 Simulation Environment and Setup

A dedicated physical layer is used for simulating packet collisions. Since we are not interested in further faults regarding the wireless communication (e.g. bit errors), no proper radio propagation model is used. Beside the integration of the self-stabilizing TDMA and CSMA protocol as explained in Sect. 4 we implemented a common CSMA-CA algorithm that is compliant to the IEEE 802.15.4 unslotted CSMA/CA protocol. In the simulation all data packets are confirmed with acknowledgments, which is a prerequisite of the self-stabilizing protocols. It is important to note that all three protocols are using a duplicate filter, which is crucial if acknowledgments are lost due to packet collisions.

The simulation settings, e.g. for transceiver, are mostly compliant to the 802.15.4 standard [1]. The packet size, containing headers and payload, and acknowledgment size is set to 100 Byte and 11 Byte respectively. The transfer rate is 250 kbit/s. The switching time of the radio is set to 0.192 ms and a long inter frame spacing (IFS) of 0.64 ms is assumed. Other than the given values

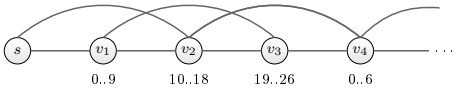
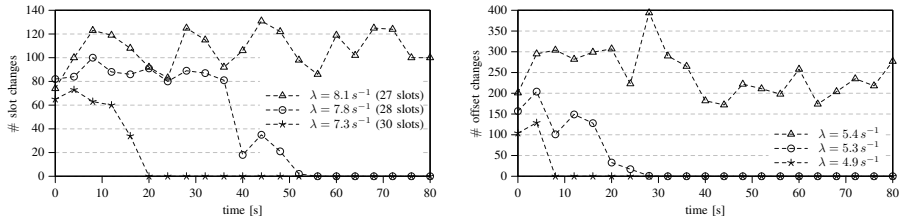


Fig. 2. Line topology and optimal slot assignment

of the 802.15.4 standard, the frame retry (is done if no acknowledgment was received) and the maximum number of backoffs (if the channel is busy) are set to infinity to avoid any kind of packet loss. The reason for this is that only reliable data transmission is considered in this paper.

5.2 Convergence Behavior in a Line-Topology

In the first simulation we investigate the ability of the two protocols to reach eventually a valid slot assignment that allows a conflict-free communication. For this purpose we simulated a line topology with one sink s and 10 nodes as well as a constant packet rate. Exactly one packet is generated by each node in each super-frame and forwarded to the sink. We considered this kind of traffic generation because it allows an easy calculation of the minimum required number of slots.



(a) Self-Stabilizing TDMA (b) Self-Stabilizing CSMA

Fig. 3. Convergence behavior in a line topology

The conflict graph for the simulated line topology is shown in Fig. 2. Note that in this experiment the communication range is set equal to the interference range. Based on this graph we determined the minimum number of TDMA slots that are required to transfer all packets without conflict to the sink. The sink needs no slot. Node v_1 needs at least 10 slots in order to transfer the own and 9 forwarded packets to the sink. Node v_2 is not allowed to reuse slots of v_1, but has itself a demand of 9 slots. Now, v_3 (demand of 8 slots) is not allowed to reuse slots of v_2 and v_1. Violating the latter condition would cause a collision, because a sending attempt of v_1 interferes with the packet transmission from v_3 to v_2. However, the demand of v_4 and the following nodes can be completely satisfied reusing slots from v_1. As a result the total number of required time-slots is 27 (see Fig. 2).

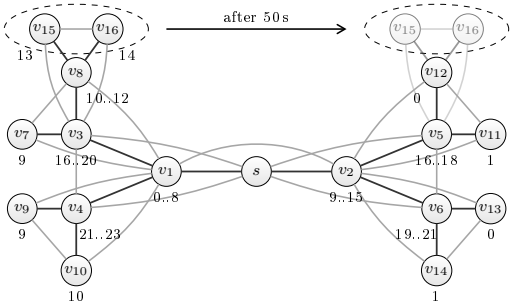


Fig. 4. Tree topology with optimal slot assignment

The self-stabilizing TDMA protocol is simulated for different numbers of time-slots until a steady state is reached. To be comparable with the self-stabilizing CSMA the packet rate λ, which depends on the number of slots and the slot length is calculated (λ is the inverse of the super-frame length). To visualize the convergence behavior we use a histogram plot over the time in which the number of slot changes are plotted. If no slot changes are necessary the protocol reaches a steady state. Representative the results of two simulation runs are depicted in Fig. 3.

The self-stabilizing CSMA doesn't converge for 27 slots, because packet collisions in the stabilization phase can never be compensated. A steady state is reached for 28 slots and $\lambda = 7.8s^{-1}$. As expected the convergence time decreases with the increase of the number of slots and decreased packet rate.

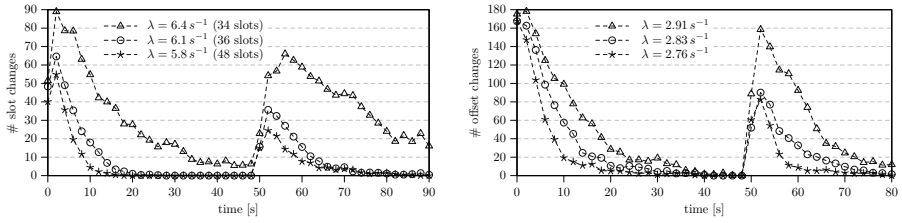
For the self-stabilizing CSMA a stabilization can be observed for $\lambda = 5.3s^{-1}$. The ratio of the maximum achievable packet rate between TDMA and CSMA is approximately $\frac{2}{3}$. This is reasonable by considering the fact that the TDMA can exploit the whole channel, whereas in case of the CSMA non-used gaps between send attempts exist (external fragmentation). By randomly choosing a valid assignment these gaps have an average size of half a slot-time which results in a channel utilization of $\frac{2}{3}$ for the given scenario.

5.3 Convergence Behavior in a Tree Topology

In the second experiment the convergence in a tree network with topology changes and additional interferences is investigated. The graph is depicted in Fig. 4. Packets are forwarded to the sink s using the bold lines. Additional signal interferences are marked as gray, thus $E_{\text{if}} \neq E$. Furthermore, packets are created in the same fashion as explained in Sect. 5.2. After 50 s the topology is changed by repositioning the nodes v_{15} and v_{16}. A valid slot assignment with the need of 24 slots is shown in the graph. The number of required slots stays constant after the topology change due to the tree's symmetry.

A total number of 100 simulations using different seeds are run and the convergence curves are averaged. The results are shown in Fig. 5. For all depicted

curves the number of slot changes successively decreases. Additionally, by reducing the traffic rate the convergence time can significantly be increased. The topology change after 50 seconds has a similar effect as starting from an initial state. This is reasonable because the required amount of slots changes for all nodes.



(a) Self-Stabilizing TDMA (b) Self-Stabilizing CSMA

Fig. 5. Averaged convergence behavior in a tree topology

The self-stabilizing TDMA shows a very slow convergence behavior for 34 slots. For this case a utilization of only 70 percent is reached in comparison to the optimal solution of 24 slots. In case of the self-stabilizing CSMA the achievable data rate is further reduced. This behavior is explainable by the more complex structure of the graph with additional interference. Also the used multi-slot allocation approach has an influence on the convergence time. Future improvements of this algorithm may significantly increase the performance of the algorithm.

Nevertheless, the simulation proves the fault-tolerance by demonstrating the ability of reacting to topology changes. Specially for the self-stabilizing TDMA protocol this an important property, since most recent TDMA protocols basically use a static slot assignment.

5.4 Data Gathering

So far a small network with uniform packet transmission has been simulated. Next the performance of the protocols in a data gathering application is examined and compared with IEEE's 802.15.4 unslotted CSMA/CA protocol, which is widely used in distributed applications.

For the simulation a connected, unit-disk graph with 400 nodes is used. The topology is created by using a quadratic grid in which all nodes are randomly repositioned around their initial position. This reflects a scenario for which nodes are more less homogeneously deployed in an area to be observed. The interference radius is twice the communication radius, which results in an average density of 5.6 neighbors and an average physical density of 23.3 neighbors. It is assumed that a clear channel assessment can not detect sending nodes in the interference circular ring. This leads to hidden nodes with a high probability. Furthermore, a

central node serves as data-sink to which all packets are forwarded. The routing itself is done by exploiting a spanning tree protocol, for which a breadth-first search is performed before the start of the simulation. Furthermore, we consider a Poisson distributed traffic generation with rate λ for all nodes.

The following settings for the three simulated algorithms have been obtained empirically for the given scenario. The number of slots for the self-stabilizing TDMA protocol is set to 100 which leads to a super-frame length of 456 ms. The same super-frame length is used by the self-stabilizing CSMA. The unslotted CSMA/CA protocol runs with a minimum backoff and maximum backoff exponent of 7 and 10 respectively. These values are much higher than the default ones in order to decrease the probability of data collisions.

We use different metrics for the evaluation of the protocols. The *1-hop delay* is the expected delay between receiving a packet and the successful transmission to the next hop. This metric is prefered to the end-to-end delay, which depends on the number of hops to a base station. To determine the throughput of the network we measure the *packet delivery rate* at the sink, which is the rate of received packets divided by the total number of nodes. Due to the fact that no packets are discarded (retransmission value is set to infinity), this metric allows to measure the maximum packet rate which can be reliably transmitted, i.e. without packet loss. The self-stabilizing MAC-protocols have the ambition to minimize packet collisions and packet loss respectively. For this purpose the *transmission fail probability* is introduced. This is the probability that either the data packet or the acknowledgment is lost and a retransmission becomes necessary. Obviously, an overloaded network results in an overflow of a node's queue. Even in cases when the network can achieve the required load, the necessary queue lengths are of importance. In the simulation the number of queued packets is logged for each node. The *required queue size* is the maximum number of packets in a queue among all nodes during the simulation.

The retrieved delivery rates for the simulation runs are depicted in Fig. 6(a). In case that the system is not overloaded the delivery rate is equal to λ. By increasing λ the maximal packet rate can be determined, for which each protocol reaches a point of saturation: the CSMA at $\lambda \approx 0.06\ s^{-1}$, the self-stabilizing CSMA at $\lambda \approx 0.08\ s^{-1}$, and the self-stabilizing TDMA at $\lambda \approx 0.13\ s^{-1}$. These results are also observable in Fig. 6(b) which shows the 1-hop delay. Furthermore, this plot reveals that the CSMA protocol achieves very small delays in comparison to the self-stabilizing protocols. The delay of the self-stabilizing CSMA is between the value of the pure CSMA and the TDMA protocol. Here the property of the self-stabilizing CSMA to behave like an ordinary CSMA for low traffic and like an TDMA protocol for high traffic is noticeable.

A high throughput is achievable by minimizing the collision probability when accessing the wireless channel, which is depicted in Fig. 6(c). The self-stabilizing TDMA as well as self-stabilizing CSMA protocol require almost no retransmissions until the saturation point is reached. Contrary to this, the transmission fail probability of the CSMA strongly rises for higher data rates.

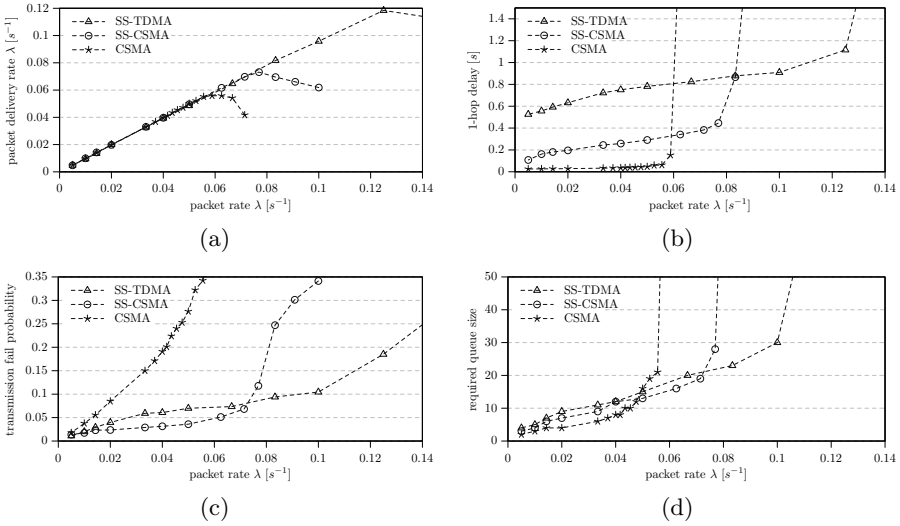


Fig. 6. Simulation results for data-gathering scenario

Finally, Fig. 6(d) shows the required queue size for the MAC-protocols. This value rises linear with the packet rate until the protocol gets in saturation. Notable is the fact that all protocols nearly need the same queue length.

It was shown that the proposed algorithms have much higher delays in comparison to the CSMA. Although the implemented self-stabilizing CSMA performs better than the TDMA, it doesn't reach the low delay of the CSMA protocol. But even more important is the fact, that the self-stabilizing protocols can cope with higher data rates and strongly avoid collisions. Although the self-stabilizing TDMA outperforms the self-stabilizing CSMA it needs a synchronization service. On the contrary the self-stabilizing CSMA runs in asynchronous mode as the pure CSMA.

6 Conclusion

In this paper we introduced a graph-theoretic model for transmission conflicts in wireless sensor networks. Based on this model the shortcoming in handling the hidden node problem of distributed TDMA approaches is shown. Afterwards we present two probabilistic self-stabilizing algorithms that eventually converges against a conflict free state: an adaptive TDMA and CSMA algorithm. Both algorithms have no additional control overhead, such as the need of exchanging controlling information between neighbors. However, for the adaptive TDMA a synchronization service is needed. The advantages of the proposed protocols are the ease of implementation with a small footprint, the high fault-tolerance in case of topology changes, the traffic awareness, and the achievable throughput. The

latter is shown in simulations, in which the algorithms outperform the 802.15.4 unslotted CSMA algorithm regarding achievable throughput.

In future research we want to implement the adaptive CSMA protocol on real hardware. Here, strategies to cope with other sources of packet loss have to be investigated. In addition an optimization of the multi-slot allocation protocol has to be performed. This may lead to a significant improvement of the performance.

References

1. IEEE Standard 802.15.4-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.
2. G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo. Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks. In *SenSys '06: Proc. 4th Int. Conf. on Embedded Networked Sensor Systems*, pages 293–306, New York, NY, USA, 2006. ACM.
3. M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *SenSys '06: Proc. 4th Int. Conf. on Embedded Networked Sensor Systems*, pages 307–320. ACM, 2006.
4. W. Crowther. A System for Broadcast Communication: Reservation-ALOHA. In *HICSS '73: Proc. 6th Hawaii Int. Conf. on Systems Sciences*, pages 371–374, 1973.
5. S. C. Ergen and P. Varaiya. TDMA Scheduling Algorithms for Wireless Sensor Networks. *Wireless Networks*, 16(1):985–997, January 2010.
6. T. Herman. Probabilistic Self-Stabilization. *Inf. Process. Lett.*, 35(2):63–67, June 1990.
7. J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proc. 2nd Int. Conf. on Embedded Networked Sensor Systems*, pages 95–107, New York, NY, USA, 2004. ACM.
8. I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Trans. Netw.*, 16(3):511–524, 2008.
9. F. Sivrikaya and B. Yener. Time Synchronization in Sensor Networks: A Survey. *IEEE Network*, 18(4):45–50, July 2004.
10. W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen. TreeMAC: Localized TDMA MAC Protocol for Real-Time High-Data-Rate Sensor Networks. *Pervasive Mob. Comput.*, 5(6):750–765, 2009.
11. M. Sun, K. Sun, and Y. Zou. Analysis and Improvement for 802.15.4 Multi-hop Network. In *CMC '09: Proc. 2009 WRI Int. Conf. on Communications and Mobile Computing*, pages 52–56, Washington, DC, USA, 2009. IEEE Computer Society.
12. A. Varga. The OMNeT++ Discrete Event Simulation System. In *ESM '2001: Proc. 15th European Simulation Multiconference*, Prague, Czech Republic, 2001.