Performance of Energy-Efficient TDMA Schemes in Data-Gathering Scenarios with Periodic Sources

Christian Renner, Volker Turau, and Christoph Weyer

***Abstract*—Energy-efficient transportation of periodical sensor readings towards a single sink in wireless sensor networks is a challenging task. In general, two data-gathering strategies exist: on-demand and bulk data forwarding. For both strategies, cross-layer techniques are a promising approach, where TDMA is tailored to the underlying routing tree. Therefore, different TDMA schemes are compared regarding achievable throughput, packet delay, and energy-efficiency for various sampling rates and scenarios. Existing schemes perform well in dedicated topologies only. The new and simple TDMA scheme presented in this paper outperforms its predecessors in all scenarios under consideration. These findings are substantiated by both theoretical analysis and extensive simulation.**

***Index Terms*—Data-gathering, energy-efficiency, sampling rate, TDMA, throughput, wireless sensor network**

I. Introduction

Monitoring of processes is required to derive models for the analysis of past and the prediction of future behavior of the environment under investigation [7]. To obtain precise models, high-resolution data-gathering is mandatory. In many application domains this end can be effectively met by wireless sensor nodes. They are small in size and thus suitable for non-invasive and high-resolution data sampling. Furthermore, they are equipped with a radio for wireless communication, hence offering the capabilities for easy and unattended collection at a data sink.

Network size, density, and the sampling rate, i.e, throughput, are the key requirements for dimensioning a data-gathering application. Yet, it is uncertain whether the desired throughput can be achieved for a given network size and density, as literature lacks practical advices. Clearly, for a fixed bandwidth of all links, the connection between the sink and the network is the main bottleneck. Yet, this is only part of the picture. As sensor nodes share the wireless channel, concurrent transmissions may lead to collisions. This problem is intensified in multi-hop networks, as nodes may be required to forward data from remote sources, so that they must act as both sender and receiver. Hence, bandwidth is shared between both tasks. In conclusion, it becomes apparent that the sink cannot receive data at full bandwidth due to collision-prevention and nodes serving as forwarders. Analyzing throughput is a complex task, that mainly depends on efficient channel utilization. In-network data compression [12], error correcting codes [10], and packet length optimization [11], [17] are valid instruments to further improve bandwidth utilization. These techniques form a self-contained area of research and are thus not considered.

In multi-hop networks with a single sink, tree-routing promises high performance [20]—i.e., high throughput and energy-efficiency as well as low packet delay. Packet loss and thus packet retransmissions are minimized by choosing parents according to link-quality metrics. Short paths allow for both energy-efficiency and an improved sampling rate, because forwarding involves less intermediate hops. Besides energy-efficiency, reliable collection of all data is required by many applications. This is achieved through automatic repeat request (ARQ) protocols [6], [18]. Congestion control is required in order to prevent buffer overflows and the loss of data [6]. This task can be effectively solved by a simple cross-layer approach in which all components are optimized for the expected traffic flow in the underlying routing tree.

Time-Division Multiple Access (TDMA) has been found to offer high performance for high-resolution data-gathering for various reasons. Most importantly, it is designed to improve the effective bandwidth and energy-efficiency by avoiding collisions under high traffic. Furthermore, TDMA is generally capable of preventing overhearing and idle listening, two severe sources of energy wastage. Application-aware schemes reinforce these advantages by setting up schedules according to an a-priori known traffic pattern. This particularly applies to periodic data-gathering in combination with tree-routing. However, TDMA comes at the cost of tight clock synchronization and extra effort for scheduling wireless transmissions.

Practical advices on network requirements for a specific data-gathering task or—expressed from the contrary perspective—the achievable throughput of a given network are rare. This paper is intended to partially fill this gap. We therefore define the underlying data-gathering scenarios and a reliable protocol. Based on this, we conduct a theoretical analysis of the optimal throughput of different TDMA schedules and verify our findings via simulation, consisting of more than $140,000$ individual simulation runs. The main contribution of this work is to give practical advices on the achievable network output and efficiency of the analyzed schedules in a network of known size and density.

The authors are with the Institute of Telematics at Hamburg University of Technology, Schwarzenbergstrasse 95, D-21073 Hamburg, Germany (e-mail: {christian.renner,turau,weyer}@tu-harburg.de).

The research presented in this paper has been partially funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under contract number TU 221/4-1.

II. Data-Gathering with Periodic Sources

In this section, we describe the wireless sensor network scenario under consideration and present the data-gathering protocol used to collect data from the network. A theoretical analysis of the maximum data sampling rate is also conducted.

A. Data-Gathering Scenario

A wireless sensor network as investigated in this paper consists of N nodes v_i, where $i = 0, \cdots, N-1$. Among these nodes there is a single data sink v_0, which differs in three ways. It does not take any measurements, has a permanent power supply, and possesses a large persistent storage. The remaining $N-1$ nodes act as periodic data sources and are equally equipped, particularly regarding their sensing devices and the radio chip with bandwidth W. These nodes are periodically sampling sensor readings. If a node has sampled d bits, it prepares one data packet of size p and stores it in a buffer. The corresponding sampling rate r_S is common to all nodes and is expressed in $\frac{\mathrm{kbit}}{\mathrm{s}}$. This leads to a network output (per time unit) of $o_N = r_\mathrm{S}\,(N-1)$, which must not exceed the achievable throughput σ at the sink.

Nodes have a limited communication range R_com, so that direct communication between an arbitrary node and the sink is not possible in general. All available communication links are given by the set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Here, a tuple $(v_i, v_j) \in \mathcal{E}$ implies that nodes v_i and v_j are within distance R_com and bidirectional communication between them is possible. Hence, a wireless sensor network is formally described as a graph $G = (\mathcal{V}, \mathcal{E})$. The density ϱ of a wireless network is defined as the average number of neighbors per node.

Many-to-one communication from each node in the network to the sink v_0 can be efficiently achieved using a routing tree $\mathcal{T}$ of G rooted in v_0. The children of a node v_i in $\mathcal{T}$ are identified by the set $\mathcal{C}_i$ and the number of leafs in the tree is denoted L. In order to make efficient use of the radio, TDMA is employed at the MAC layer, as proposed by Estrin *et al.* [9]. Time is divided into rounds and each round is subdivided into a set $\mathcal{S}$ of R slots of equal length T_s. The assignment of subsets $\mathcal{S}_i \subseteq \mathcal{S}$ to the v_i is called schedule.

B. Data-Gathering Protocol

Tree construction has been widely explored in the literature [20], so that we do not consider this aspect here. However, wireless links vary over time and may become interrupted [19]; hence, tree maintenance of some kind is required. In fact, there are different approaches on this matter. Firstly, it is possible to apply local repair and use multiple parents per node, as employed in Dozer [3]. A different solution is tree reconstruction, initiated by the sink either periodically or if throughput drops below a critical limit. The main advantage of tree reconstruction is the static nature of the corresponding trees, yielding a predictable traffic pattern. The latter can be exploited to construct efficient schedules. If, in contrast, continuous tree maintenance is applied, sending schedules become inefficient, as the traffic pattern changes. Due to the existence of local knowledge only, rescheduling may not be viable. Provided with these findings, we assume periodic tree reconstruction and analyze network performance with static, precalculated trees in this paper. In this context we suggest that slot distribution is performed during tree reconstruction. A practical approach would be as follows: The sink initiates tree reconstruction by exploiting the current tree and all nodes switch on their transceivers subsequently. The information required for tree reconstruction is exchanged and slots are distributed afterwards. Messages sent during this phase can be used for clock synchronization. We do not enforce a particular medium access in this period; e.g., using CSMA would be an appropriate solution.

In close relation to tree maintenance issues, there exist two orthogonal data-collection strategies. The on-demand strategy enables nodes to forward foreign packets and send own ones quickly [3]. Therefore, low packet delay is experienced at the sink, so that this approach is to be favored, if low delay is of importance. A drawback yet is that nodes have to always listen in the slots of their children, wasting energy if nothing is to be sent. To attenuate this, a parent node switches off its transceiver, if packet reception in a child's slot does not commence before a timeout occurs. The second strategy is a cyclic two-phase system [21]. In the sensing phase, nodes switch of their transceivers and sample data. In the forwarding phase, each node also forwards all packets previously generated in its subtree. Note that nodes do not forward packets generated in the current forwarding phase. As soon as a node detects the last packet, it transits back to the sensing phase. This approach promises efficient usage of the radio, but comes at some cost. Packet delay is increased, and buffers must be large enough to store packets created during a sensing phase. Moreover, the achievable sampling rate is reduced, as only part of the time is used for forwarding. Our protocol has been designed to support both strategies, allowing for a quantitative comparison.

The unstable nature of wireless links causes packet loss [26]. This can have a negative effect on the application itself. Hence, ARQ must be used in order to assure successful data delivery, i.e., reliability. As reported in [18], hop-to-hop acknowledgments are to be favored here. Hence, we adapt this technique and resend each packet, until a corresponding acknowledgment is received. Nodes store created and received packets in a buffer, that is, e.g., located in the EEPROM of a node. This ensures reliable data storage. To avoid buffer overflows, a waiting mechanism is used. If a node receives a packet from one of its children, it advises this node to skip sending slots by adding extra information to the acknowledgment. To make this mechanism work even in case a buffer overflow has already occurred, acknowledgments carry a flag signaling whether the just received packet could be stored in the buffer. In order to gain fast feedback, acknowledgments are sent immediately after the reception of a data packet—even if the latter cannot be stored in the buffer—and within the same slot. This task can only be accomplished by a cross-layer approach, in which the MAC layer is aware of the application behavior, i.e., the packet store-and-forward strategy. The sender removes a packet from its buffer only upon receiving a positive

ACK. As a result, it is sufficient to choose T_s so that one data packet and the corresponding acknowledgment can be sent with given bandwidth W, plus a short amount of time to compensate for propagation delay and synchronization errors. Note that piggy-backing clock information in ACKs can be used to enforce clock synchronization.

C. Optimal Sampling Rate

As noted in Sect. I, maximizing the network output o_N is of practical interest in a data-gathering scenario. Duarte-Melo and Liu have shown that in case of a single sink, the maximum sampling rate r_S per node is in $\Theta(\frac{W}{N})$ [5]. The required slot length T_s is obtained by dividing the number of bits to be transmitted in one slot by the available bandwidth of the radio transceiver. In general, this number of bits is proportional to d. It follows that $r_\mathrm{S} \in \Theta(\frac{d}{NT_\mathrm{s}})$, which is a more natural representation in context of TDMA. In the following, we will develop a corresponding upper bound of r_S for the protocol presented in Sect. II-B.

The maximum o_N is bounded by the theoretical throughput σ at the sink. Given the slot length T_s, the ratio η of slots assigned to the sink's children $\mathcal{C}_0$, and R, we find

$$o_N \leq \sigma = \eta \frac{d}{T_\mathrm{s}} \quad \text{where} \quad \eta = \frac{\sum_{v_i \in \mathcal{C}_0} |\mathcal{S}_i|}{R} \ . \tag{1}$$

The sampling rate per node r_S is therefore described by

$$r_\mathrm{S} \leq \frac{o_N}{N-1} \leq \frac{\eta d}{(N-1)\, T_\mathrm{s}} \ . \tag{2}$$

The objective is to maximize o_N and r_S respectively. As d is fixed and T_s is determined by the available bandwidth, the maximization problem is equivalent to maximizing η under the following constraint:

$$\forall v_i \in \mathcal{V} \setminus \{v_0\} : \mathcal{S}_i \neq \emptyset \wedge |\mathcal{S}_i| \leq 1 + \sum_{v_j \in \mathcal{C}_i} |\mathcal{S}_j| \ . \tag{3}$$

Consequently, each leaf is assigned a single slot and can hence send (at most) one packet per round, so that $r_\mathrm{S} \leq \frac{d}{R}$. A node having among its children only leafs must not have more slots at its disposal than it can use. Carrying out the recursion makes the closure.

In a cyclic two-phase system, the lengths of a sampling phase T_spl and a forwarding phase T_fwd form the ratio

$$\frac{T_\mathrm{fwd}}{T_\mathrm{fwd} + T_\mathrm{spl}} \leq 1 \ , \tag{4}$$

that must be inserted as a factor on the right-hand side of (2).

Note that the given estimation of σ is only an upper bound, as packet loss and individual node load—inferred by the shape of the routing tree—are not considered. It is an open question how close to σ the effective throughput σ_eff can be.

III. Slot Assignment

As discussed in Sect. II, the sending schedule used for data-gathering is crucial for the performance, in particular the maximum throughput σ. Different TDMA scheduling schemes have been proposed in the literature. However, having a static routing tree and a predictable traffic pattern, static schedules are preferable for the described data-gathering scenario. This holds, because dynamic slot assignment leads to overhead and is generally suspect to collisions. Furthermore, static schedules increase energy-efficiency, as slots can be assigned during tree construction. Note that the term static implies that each node is assigned a fixed subset of slots and slots are only reassigned after tree construction. In the following, three particular schemes are introduced and discussed, based on the findings reported by Renner [15].

A. Minimum Round Length Schedules

Traditional TDMA schemes concentrate on minimizing the round length by assigning one slot to each node v_i in the network ($|\mathcal{S}_i| = 1$) and spatially reusing slots to increase each node's share of time usable for packet transmission. The frequently advertised advantage of this assignment strategy is a small value for R, which is generally proportional to the network density ϱ and is independent of N in large networks. Hence, (1) and (2) simplify to

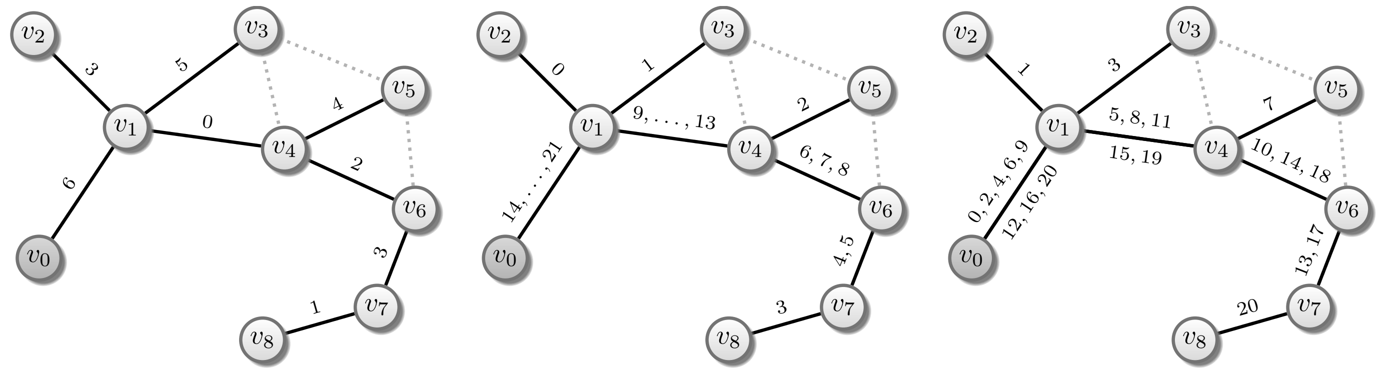
$$\eta = \frac{|\mathcal{C}_0|}{R} \sim \frac{|\mathcal{C}_0|}{\varrho} \quad \Rightarrow \quad r_\mathrm{S} \lesssim \frac{d\, |\mathcal{C}_0|}{(N-1)\, T_\mathrm{s} \varrho} \ . \tag{5}$$

Note that the corresponding estimation of σ is only valid in the case of equally sized subtrees. If this assumption does not hold, the largest subtree dominates the achievable throughput.

Assigning a minimum number of slots is a $\mathcal{NP}$-complete problem, as this problem generally involves some variant of graph coloring, i.e., reusing slots spatially after at least k hops. This require the exchange of the k-hop neighborhood information. The Color Constraint Heuristic [2] is a solution that offers smaller round length than the well-known DRAND [16]. CCH introduces a metric to decide the order in which nodes pick a color. It is based on computing a weighted sum of already colored one-hop and two-hop neighbors for each node. The node with the highest value of the metric picks a color and the procedure is repeated. Besides this centralistic approach, a decentralized algorithm has been proposed. The metric is scaled by the inverse of the weighted sum of a node's one-hop and two-hop neighbors. If this fraction is larger than a pre-defined threshold, the corresponding node picks a color.

All coloring approaches produce collision-afflicted schedules [8], most likely causing temporary forwarding stalls in the routing tree. Due to this, interference-aware schemes have been proposed, such as the radio interference detection (RID) protocol [27]. Here, the sending schedule is determined by taking perceived signal strengths and thus interference information into account. Yet, sharing this information in a k-hop neighborhood is still required. Since radio signal strength is subject to change and cannot be measured precisely, this kind of scheduling may still produce collision-afflicted schedules.

In a data-gathering application with high traffic, i.e., nodes use their single sending slot most of the time, state-of-the-art coloring schemes of this kind have been found to severely suffer from collisions [8]. Hence, we modified the CCH assignment by taking interference ranges into consideration, so that schedules



(a) CCH: $\eta = \frac{1}{7} = 0.14\ldots$ (b) LACF: $\eta = \frac{8}{22} = 0.36\ldots$ (c) SPR^{+}, $\kappa = 4$: $\eta = \frac{8}{21} = 0.38\ldots$

Fig. 1. TDMA slot assignments and corresponding slot utilization η

are almost collision-free while a small R is conserved. An example slot assignment is depicted in Fig. 1a. It reveals the main drawback of assigning precisely one slot to each node: The inherent traffic pattern of a data-gathering scenario is neglected, e.g., v_1 and v_2 have the same amount of slots for sending, although the former has to forward data from the whole network, whereas the latter must only send its own data. The details of slot assignment are of no relevance for comparison, CCH was chosen as a representative of minimum round length schedules.

B. Load-Aware Collision-Free Scheduling

A natural way to assign slots satisfying (3) is that proposed in previous work [22], which we will refer to as LACF (load-aware, collision-free). The scheme assigns to each node v_i exactly as many slots as its children have in total plus one slot for the node itself, i.e., $|\mathcal{S}_i| = 1 + \sum_{v_j \in \mathcal{C}_i} |\mathcal{S}_j|$. This procedure is equivalent to assigning for every node v_i one additional slot to each link on the path from v_i to v_0. To simplify slot assignment as well as storage and to prevent collisions, the $\mathcal{S}_i$ are disjoint and slots within one set are consecutive. To decrease packet delay, slots are ordered ascendingly from leafs to the sink. Slot distribution can be performed via a depth-first-search after tree construction. An example slot assignment is shown in Fig. 1b.

This procedure has two advantages: Collisions are completely prevented, as slots are not reused, and the traffic pattern of a routing tree is followed. Thus, one packet from each node can be collected by the sink per round. As v_0 gains knowledge about R after a completed slot assignment, it can calculate the optimal r_S and distribute this information to all nodes.

Yet, the lack of reusing slots imposes a severe drawback. R increases rapidly with growing N and also depends on the shape—mainly the depth—of the tree. In particular, R is the product of N and the average depth of all nodes in a tree. Simulation reveals that the depth increases logarithmically with N in most cases, leading to $R \sim N \log N$. Since $N-1$ slots are available at the sink for receiving data from its children, we find

$$\eta = \frac{N-1}{R} \sim \frac{1}{\log N} \quad \Rightarrow \quad r_\mathrm{S} \le \frac{d}{T_\mathrm{s} R} \sim \frac{d}{T_\mathrm{s} N \log N}\,. \quad (6)$$

As the slots within one set $\mathcal{S}_i$ are consecutive, the buffer of a node v_i must store $|\mathcal{S}_i|$ packets. Therefore, this scheme is appropriate in small networks only.

C. Spatial Path-Based Reuse

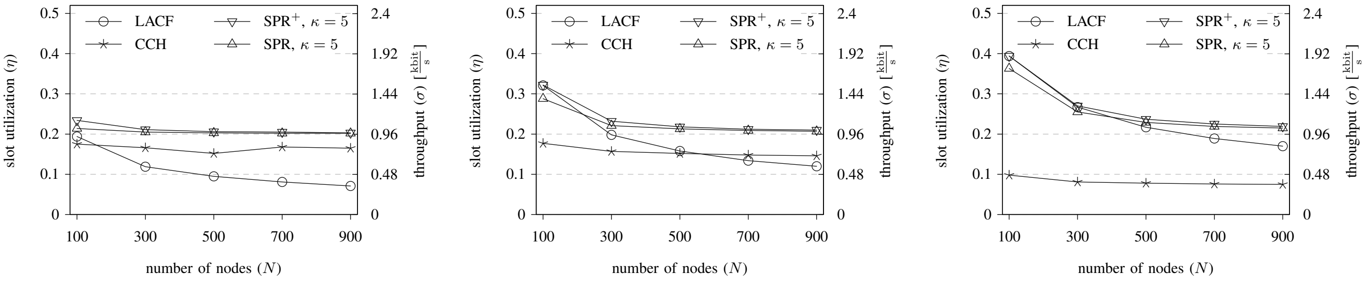
To overcome the problems of the afore-presented scheduling approaches, we have developed the spatial path-based reuse (SPR) scheme [23]. It has been proven to be very efficient in large data-gathering networks [15]. This paper further improves the performance with an extension of SPR named SPR^{+}.

The strategy behind SPR^{+} is as follows. For each path from a node to the sink, consecutive slots are assigned in ascending order towards the sink. Slots on paths with length exceeding κ are reused every κ hops so as to reduce the overall number of slots (see Fig. 1c). Note that slot utilization η is increased compared to CCH and LACF.

Like LACF, SPR^{+} provides each node with just as many slots as there are nodes in its subtree. Thus, each node is equipped with a balanced number of slots for receiving data from its children and for sending to its parent. This reflects the traffic pattern in the scenarios under consideration. In contrast to LACF, slots are assigned in a way that sending and receiving are alternating, so that buffer usage is minimized.

Although nodes are assigned a number of slots according to their subtree size, there is no need to explicitly store these slots. In order to assign precisely k slots to paths with length $k < \kappa$, every node v_i calculates a displacement vector $\mathbf{d}_i$ with κ elements. $\mathbf{d}_i[k]$ is the number of nodes in v_i's subtree (including v_i) with depth $k < \kappa$ in $\mathcal{T}$. For $k = \kappa$ the vector denotes the number of nodes in the subtree with depth κ or higher in $\mathcal{T}$. In addition, each v_i stores an offset vector $\mathbf{o}_i$, where the k'th element represents the smallest of all slots belonging to a path through v_i with displacement k. This information together with the round length and the depth h_i of v_i is sufficient to compute the set of slots assigned to v_i:

$$\mathcal{S}_i = \Big\{ \; s \;\mid\; 1 \le k \le \kappa,\; 0 \le d < \mathbf{d}_i[k] : \\ s = \mathbf{o}_i[k] + k\,d + (-h_i) \bmod k \; \Big\} \quad (7)$$



(a) $\varrho = 6$ (b) $\varrho = 12$ (c) $\varrho = 24$

Fig. 2. Slot utilization and theoretical throughput for trees with $|\mathcal{C}_i| \leq 8$

In comparison with LACF, R is reduced, because slots may be reused on paths with node depths larger than κ. Although R is still large, high throughput is preserved, since R is bounded by $N\kappa$. The maximum sampling rate r_S of SPR$^+$ can be estimated as follows:

$$\eta > \frac{N-1}{(N-1)\kappa} = \frac{1}{\kappa} \quad \overset{R\approx L\kappa}{\Longrightarrow} \quad r_\mathrm{S} \lesssim \frac{d}{(N-1)\,T_\mathrm{s}\kappa}. \qquad (8)$$

In large networks, this yields a good estimation of r_S, whereas R will be considerably lower in small networks, yielding a higher r_S.

Inter-path collisions are prevented by assigning disjoint sets of slots to the individual paths. Thus, collisions are possible exclusively on the same path and are likely in sparse networks only, where paths bend around so-called voids. κ must be chosen carefully in context of ϱ and R_com. Simulation has shown that $\kappa = 5$ or 6 is sufficient in most cases.

From (8) it is evident that SPR$^+$ allows for a higher r_S than LACF particularly in large networks, as the factor $\log N$ in (6) is replaced by the constant κ. Moreover, (8) already approaches the minimum found by Duarte-Melo up to a small constant factor. In small networks however, slots are rarely reused and SPR$^+$ and LACF offer a similar r_S.

IV. Evaluation

To verify our theoretical models and to compare the different schemes, extensive simulation using the ns-2 framework from [15] has been performed. The simulation setup is first described in this section. It follows the presentation and discussion of the results, encompassing throughput, packet delay, and energy-efficiency in the context of comparing on-demand and two-phase data collection.

A. Simulation Settings

For all simulations, the two-ray ground propagation model is used. Therefore, packet loss caused by changing signal strengths is excluded. This eliminates one factor influencing the results, so that a smaller amount of simulation runs is required. However, the two-ray ground model leads to different SINR levels depending on the distances between all simultaneous senders and the corresponding receiver(s). Collisions are simulated by requiring a SINR of at least 10 dB. A lower SINR leads to packet loss on the receiving side.

All nodes have a communication range of $R_\mathrm{com} = 40$ m and a bandwidth of $W = 19.2\ \frac{\text{kbit}}{\text{s}}$. All topologies, routing trees, and slot assignments have been precalculated using dedicated tools, so that the same settings have been used for each different slot assignment algorithm. Hence, all results have been obtained under equal conditions.

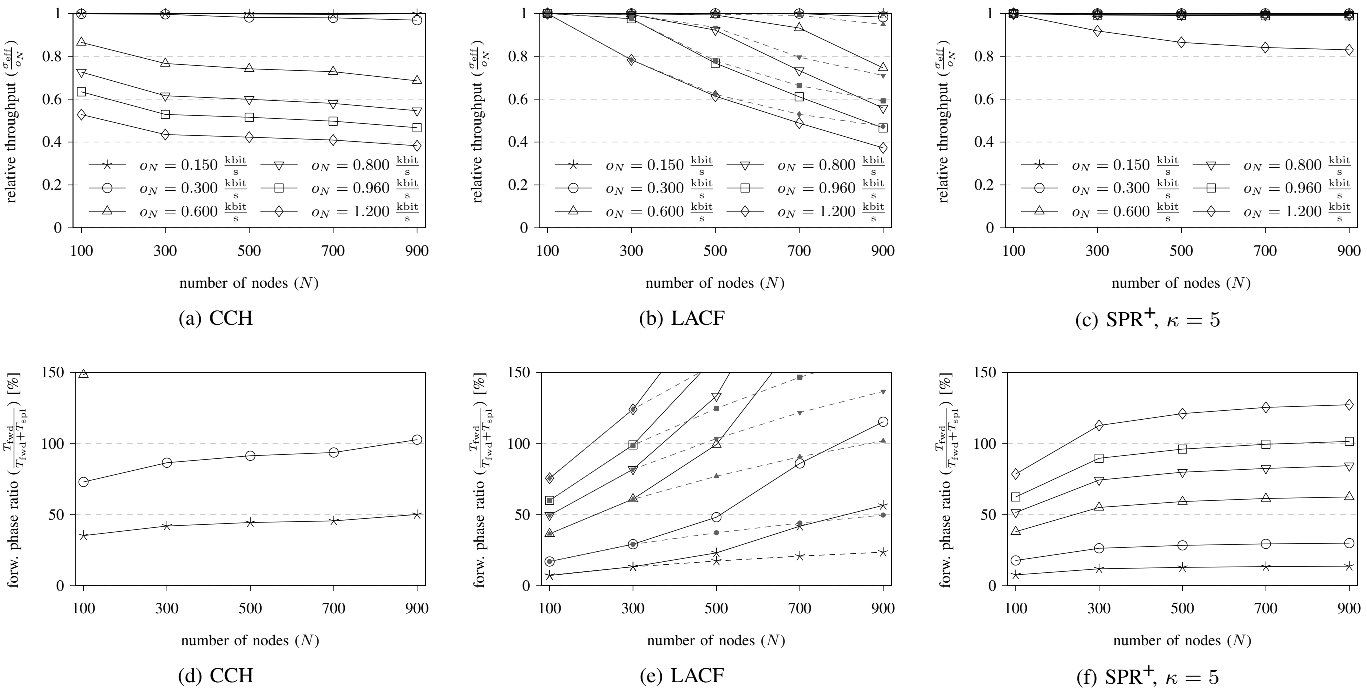
Topologies of densities $\varrho = 6, 9, 12, 18, 24$ and values of N from 20 to 900 in a rectangular area are used. For each combination of these parameters, 50 topologies have been generated with the sink located in the center of the network. Routing trees have been built by a breadth-first search. Particularly in dense networks, the maximum number of children per node influences performance, so that we have constructed for each topology one tree without limiting the number of children and one with at most 8 children. The latter is a trade-off between ensuring connectivity and saving available memory. In total, $140,000$ individual simulation runs have been performed.

Buffers are restricted to hold at most 200 packets, which is valid for a buffer located in the EEPROM. At the beginning of each experiment, every node in the network has an empty buffer. In each simulation run, every node generates 40 packets with constant sampling rate $r_\mathrm{S} = o_N/(N-1)$ and $d = 240$ bits. The forwarding phase of the two-phase collection strategy starts at $t = 40d/r_\mathrm{S}$. Packet length and bandwidth lead to a slot length T_s of 50 ms.

B. Slot Utilization and Theoretical Throughput

To determine the maximum σ_eff as introduced in Sect. II-C, η and σ are computed for all schemes. Based on these results we will derive a range of values for o_N to be analyzed in the simulations. The average values of η and σ for all generated topologies are displayed in Fig. 2. The diagrams support the findings from Sect. III. As foreseen in (6), η of LACF drops heavily with increasing N. Slot reusing of SPR$^+$ gives relief to this. The influence of κ is effective in large networks only, as paths are too short in small networks and slots are not reused. Note that η of CCH is independent of N. The slightly higher values for $N = 100$ are due to densities below specification, caused by a weakness of the topology generator.

As stated in (5), slot utilization of CCH is proportional to ϱ. In sparse networks with $\varrho = 6$ (see Fig. 2a) CCH and SPR exhibit comparable throughput, whereas LACF is falling



(a) CCH

(b) LACF

(c) SPR$^+$, $\kappa = 5$

(d) CCH

(e) LACF

(f) SPR$^+$, $\kappa = 5$

Fig. 3. Relative throughput of on-demand (upper row), share of forwarding phase of two-phase data collection, $|\mathcal{C}_i| \leq 8$, $\varrho = 12$

considerably short, as trees have huge depths. While CCH is rather competitive with LACF for $\varrho = 12$, Fig. 2c indicates that it is inappropriate in dense networks and even inferior to LACF for large N. It comes as no surprise that LACF and SPR$^+$ are affected by ϱ, too. In small networks, slots are rarely reused and a higher ϱ leads to trees with lower depth, resulting in a smaller R. For large N, most leafs (or nodes considering SPR$^+$) have a depth larger than κ, so that depth becomes insignificant, as slots are frequently reused.

These findings are generally valid for trees without a restricted number of children per node with the exception of CCH. In dense networks, theoretical throughput σ appears large, as the sink has many children. Yet, trees are unbalanced and some subtrees just consist of a few nodes, so that σ is inappropriate to forecast σ_{eff} and maximum o_N, respectively.

In conclusion, the results of Fig. 2 in combination with (1) give rise to a network output o_N in the range of 0.15 to 1.2 $\frac{\text{kbit}}{\text{s}}$ for the following simulations.

C. Effective Throughput

The effective throughput σ_{eff} is the ratio of packets collected by the sink during $40d(N-1)/o_N$ (cf. Sect. IV-A). Figure 3 (upper row) displays the relative throughput $\frac{\sigma_{\text{eff}}}{o_N}$. Values below 1 indicate that the network does not support this o_N.

CCH can merely operate the lowest o_N (cf. Fig 3a), which is due to unbalanced routing trees. Nodes close to the sink suffer from buffer congestion. As round length is proportional to ϱ, effective throughput is slightly higher in sparse networks.

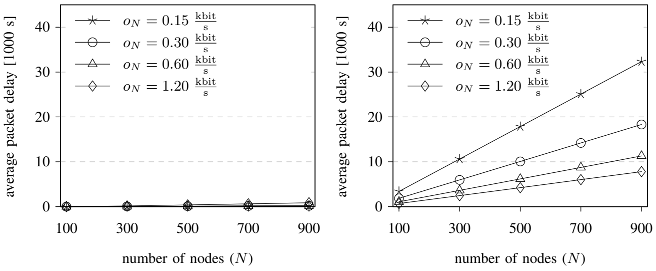
LACF can operate all sampling rates for $N = 100$, as depicted in Fig. 3b. With increasing N, σ_{eff} is rapidly decreasing, as round length grows. For $N \geq 300$, the number of slots assigned to nodes close to the sink exceeds their buffer size, so that more and more slots are wasted. The theoretical performance of LACF, i.e., without buffer limitations, is indicated by dashed curves. Although σ_{eff} is increased for larger N, it is still below o_N. In dense networks, LACF is able to achieve larger values of o_N, because average node depth in $\mathcal{T}$ becomes smaller and thus less slots are generated. Particularly in dense networks, o_N can be slightly raised, if $|\mathcal{C}_i|$ is not restricted.

For SPR$^+$, network density has an impact on σ_{eff} only if N is small, as slots are merely reused in this case. Thus, SPR$^+$ behaves like LACF here. Not restricting the number of children allows for a slightly higher sampling rate due to a lower R. These findings are also valid for SPR.

In conclusion, it shows that SPR$^+$ achieves the theoretical throughput derived in Sect. IV-B. The main result is that the sink can precisely calculate the maximum sampling rate by means of (8) once R is known, i.e., after slot assignment. This is also true for LACF, but (effective) throughput is lower for this scheme. Sampling rate prediction for CCH is also possible, but the size of the largest subtree must be known, as (5) fails. Yet, CCH can operate at low sampling rates only.

D. Forwarding Time

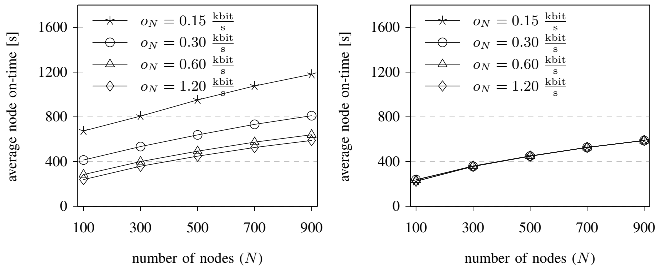
In a two-phase forwarding strategy, the share of time occupied by data collection is of interest. The smaller it is, the more efficient the radio can be used. Figure 3 (lower row) displays this ratio. Note that values above 100% indicate that



(a) on-demand

(b) two-phase

Fig. 4. Average packet delay of SPR$^+$ with $\kappa = 5$, $|\mathcal{C}_i| \leq 8$, and $\varrho = 12$



(a) on-demand

(b) two-phase

Fig. 5. Energy-efficiency of SPR$^+$ with $\kappa = 5$, $|\mathcal{C}_i| \leq 8$, and $\varrho = 12$

two-phase data collection is impossible at the corresponding network output o_N.

According to the inability of CCH to collect data even at low values of o_N, two-phase data collection is generally infeasible as supported by Fig. 3d. LACF, see Fig. 3e, cannot be applied in large networks as discussed earlier due to buffering issues, as discussed earlier. As SPR$^+$ is free from this problem, even high data rates are accomplished. Due to slot reuse, the share of the forwarding phase is almost unaffected by a large N. Network density has analogous impact on all results as on the effective throughput in the previous section.

E. Packet Delay

The preceding findings show that SPR$^+$ offers best performance for both on-demand and two-phase data collection. Hence, average packet delay is exclusively examined for this scheme. Figure 4a indicates that packet delay is increasing with growing N, as average node depth increases and packets of additional nodes arrive at the sink after an increased number of hops. In contrast, o_N has no influence as long as $\sigma_{\text{eff}} = o_N$. This behavior is traceable, because this equation guarantees that all packets can be collected in one round.

Packet delay is considerably larger for two-phase data collection, as all packets have an inevitable delay of $\frac{T_{\text{fwd}}+T_{\text{spl}}}{2}$ plus the time needed for actual collection. The larger o_N, the shorter the time between two forwarding phases and thus packet delay becomes smaller (cf. Sect. IV-A). This effect is clearly visible in Fig. 4b. Packet delay grows linearly with N for a constant o_N, because η stays approximately constant for SPR$^+$ while the number of packets increases linearly.

F. Energy-Efficiency

Simulation results show that SPR$^+$ and LACF produce the highest energy-efficiency among the examined TDMA schemes. Here, energy consumption is measured as the total time of all nodes spending on: listening on, receiving from, and sending over the wireless channel. This is a valid metric, as these actions require approximately the same power. In the case of on-demand forwarding, see Fig. 5a, energy consumption is higher for smaller values of o_N. This is caused by a larger number of slots not being used for forwarding. If o_N is approaching the maximum possible σ_{eff}, both forwarding strategies achieve similar energy-efficiency as shown by Fig. 5b, because in this case we find $\frac{T_{\text{fwd}}}{T_{\text{fwd}}+T_{\text{spl}}} \to 1$. However, energy consumption is not influenced by the sampling rate in a two-phase strategy, since collection time only depends on the number of packets to be collected. Note that energy-efficiency is, in all cases, proportional to the average depth of all nodes in $\mathcal{T}$, which is a lower bound on energy consumption.

V. Related Work

Many-to-one data-gathering has been widely studied in the literature. Yet, practical advices on network dimensioning for achieving a required r_{S} are rare. Duarte-Melo and Liu show that the maximum sampling rate r_{S} in data-gathering network with a single sink is in $\Theta(\frac{W}{N})$ [5]. They also discuss trade-offs between increasing sampling rates and reducing energy consumption. Their work is a fundamental basis for the findings reported in this paper. Cui *et al.* analyze energy-efficiency and packet delay [4]. They show that rate-adaptive scheduling can be used to trade off energy consumption against packet delay. A traffic adaptive protocol is introduced in [24]. Its main focus is on energy-efficiency and the support of different sampling rates among nodes. However, particular advices on choosing r_{S} are discussed in neither contribution.

Throughput, delay, and energy-efficiency have been frequently addressed in the literature. Due to the variety and amount of proposed protocols, we can only point out a few important contributions being tailored to the needs of data-gathering with a single sink. A vast body of CSMA-based protocols has been proposed. T-MAC is a generic solution that supports data-gathering by adaption to network load [25]. Another early and well-known work is D-MAC [13]. Unlike T-MAC, D-MAC exploits the inherent traffic pattern induced by a routing tree. Although it is CSMA-based, nodes use sending schedules in order to avoid collisions and to decrease latency by staggering sending slots according to node depth in the tree. While latency is low, it has been shown that collisions and idle listening negatively affect energy-efficiency [14].

The Funneling-MAC is a hybrid approach [1]. Nodes close to the sink employ TDMA, since this area is exposed to high traffic. Nodes farther apart from the sink operate CSMA in order to decrease latency. As a result, nodes at the edge of both zones must use both MACs, which is a complicated task and leads to high energy consumption. TDMA-EC is another approach to energy-efficient and collision-free data-gathering [14]. Schedules are set up according to nodes' depth in the routing tree to decrease packet delay. Yet, energy is wasted, as nodes may randomly select slots, if required.

VI. Conclusion

Different TDMA schemes have been compared for data-gathering scenarios with periodic data sources. We have shown that SPR$^+$ achieves the highest network output o_N among all examined schemes and in all scenarios. Furthermore, the construction of SPR$^+$ is simple due to employing the existing data-gathering tree. For densities between 6 and 24 neighbors per node, o_N is as large as 40% of the net link bandwidth in networks with 100 nodes and 20% in networks with 900 nodes. Another advantage of the scheme is that the maximum possible o_N is easily predictable after slot assignment. Other approaches do either not allow for precise prediction of the maximum o_N or only achieve low throughput. We have also shown that exclusively minimizing the round length is not sufficient for increasing o_N. Moreover, maximum o_N of SPR$^+$ is independent of N in large networks and is thus proportional to the theoretical maximum documented in the literature. Another important outcome of this paper is the supply of practical information on how to dimension a data-gathering network for achieving a specific r_{S}.

Two-phase data collection achieves a lower network output and thus sampling rate than its on-demand counterpart, because only a fraction of time is used for data forwarding. This results in an increased packet delay. In contrast to this, it is proven that two-phase data collection offers higher energy-efficiency. Energy savings as compared to on-demand data collection are particularly large for low sampling rates. The reason for this is that on-demand data collection is wasting energy, since low sampling rates lead to unused slots and therefore produce idle listening. This is prevented by the two-phase strategy almost completely, since nodes are listening only during the bulk forwarding phase and achieve a nearly optimal slot usage. E.g., in case of a network output of $0.150\ \frac{\mathrm{kbit}}{\mathrm{s}}$, two-phase data collection reduces energy consumption by more than 50%. For both strategies, using cross-layer techniques by tailoring TDMA to the underlying routing tree, reduces energy consumption and increases network output.

References

[1] 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 *Proc. 4th Intl. Conf. on Embedded Networked Sensor Systems*, 2006.

[2] K. L. Bryan, T. Ren, L. DiPippo, T. Henry, and V. Fay-Wolfe. Towards Optimal TDMA Frame Size in Wireless Sensor Networks. Technical report, Dept. of Computer Science and Statistics, University of Rhode Island, Kingston, USA, 2007.

[3] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: Ultra-Low Power Data Gathering in Sensor Networks. In *Proc. 6th Intl. Conf. on Information Processing in Sensor Networks*, 2007.

[4] S. Cui, R. Madan, A. Goldsmith, and S. Lall. Energy-Delay Tradeoffs for Data Collection in TDMA-Based Sensor Networks. In *Proc. 40th Intl. Conf. on Communications*, 2005.

[5] E. J. Duarte-Melo and M. Liu. Data-Gathering Wireless Sensor Networks: Organization and Capacity. *Computer Networks*, 43, 2003.

[6] C.-T. Ee and R. Bajcsy. Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In *Proc. 2nd Intl. Conf. on Embedded Networked Sensor Systems*, 2004.

[7] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, 1(1), 2002.

[8] J. Grönkvist. Comparison Between Scheduling Models for Spatial Reuse TDMA. In *Proc. 2nd Workshop on Affordable Wireless Services and Infrastructure*, 2004.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. 6th Intl. Conf. on Mobile Computing and Networking*, 2000.

[10] J. Jeong and C.-T. Ee. Forward Error Correction in Sensor Networks. In *Proc. Intl. Workshop on Wireless Sensor Networks*, 2007.

[11] I. Joe. Optimal Packet Length with Energy Efficiency for Wireless Sensor Networks. In *Proc. Intl. Symp. on Circuits and Systems*, 2005.

[12] N. Kimura and S. Latifi. A Survey on Data Compression in Wireless Sensor Networks. In *Proc. Intl. Conf. on Inf. Tech.: Coding and Computing*, 2005.

[13] G. Lu, B. Krishnamachari, and C. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks. In *Proc. Intl. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2004.

[14] B. Ren, J. Xiao, J. Ma, and S. Cheng. An Energy-Conserving and Collision-Free MAC Protocol Based on TDMA for Wireless Sensor Networks. In *Proc. Intl. Conf. on Mobile Ad-Hoc and Sensor Networks*, 2005.

[15] C. Renner. Energy-Efficient TDMA Schedules for Data-Gathering in Wireless Sensor Networks. Master's thesis, Hamburg University of Technology, 2008.

[16] I. Rhee, A. Warrier, J. Min, and L. Xu. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks. In *Proc. 7th Intl. Symp. on Mobile Ad Hoc Networking and Computing*, 2006.

[17] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin. Energy Efficiency Based Packet Size Optimization in Wireless Sensor Networks. In *Proc. 1st Intl. Workshop on Sensor Network Protocols and Applications (SNPA '03)*, 2003.

[18] F. Stann and J. Heidemann. RMST: Reliable Data Transport in Sensor Networks. In *Proc. 1st Intl. Workshop on Sensor Net Protocols and Applications*, 2003.

[19] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a Sensor Network Expedition. In *Proc. 1st Eur. Conf. on Wireless Sensor Networks*, 2004.

[20] N. Thepvilojanapong, Y. Tobe, and K. Sezaki. On the Construction of Efficient Data Gathering Tree in Wireless Sensor Networks. In *Proc. Intl. Symp. on Circuits and Systems*, 2005.

[21] V. Turau and C. Weyer. Scheduling Transmission of Bulk Data in Sensor Networks Using a Dynamic TDMA Protocol. In *Proc. Intl. Workshop on Data Intensive Sensor Networks*, 2007.

[22] V. Turau and C. Weyer. TDMA-Schemes for Tree-Routing in Data Intensive Wireless Sensor Networks. In *Proc. 1st Intl. Workshop on Protocols and Algorithms for Reliable and Data Intensive Sensor Networks*, 2007.

[23] V. Turau, C. Weyer, and C. Renner. Efficient Slot Assigment for the Many-to-One Routing Pattern in Sensor Networks. In *Proc. Intl. Workshop on Sensor Network Engineering*, 2008.

[24] M. Vahabi, M. Rasid, R. Abdullah, and M. Ghazvini. Adaptive Data Collection Algorithm for Wireless Sensor Networks. *IJCSNS*, 8(6), 2008.

[25] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems*, 2003.

[26] M. Zúñiga Zamalloa and B. Krishnamachari. An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links. *ACM TOSN*, 3(2), 2007.

[27] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher. RID: Radio Interference Detection in Wireless Sensor Networks. In *Proc. 24th Joint Conf. IEEE Computer and Communications Societies*, 2005.