On the TDMA Design Problem Under Real-Time Constraints in Wireless Sensor Networks

Technical Report 359/2007

Nicos Gollan, Jens Schmitt

DISCO | Distributed Computer Systems Lab
TU Kaiserslautern, Computer Science Department

Abstract. Many wireless sensor networks (WSNs) are used to collect and aggregate data from potentially hostile environments. Catering to this, early application scenarios did not put tight constraints on performance properties like delay, but rather focused on ruggedness and energy conservation. Yet, there is a growing number of scenarios like e.g. production monitoring, intrusion detection, or health care systems which depend on the sensor network to provide performance guarantees in order to be able to act upon the phenomena being sensed in a timely fashion. Nevertheless, these applications still face the traditional issue of energy-efficiency. In this paper, we present means to find energy-efficient medium assignments in time-slotted multi-hop networks that satisfy given real-time constraints. Specifically, we present a way to find the optimal length of time slots and periods in TDMA schemes. We also present a software to compute those values for typical sink-tree WSNs.

1 Introduction

Wireless sensor networks have requirements and characteristics that are considerably different from those of common computer networks. Issues include low energy reserves, limited processing power, uncontrolled environments and other adverse factors. The main attention has been put on meeting these restrictions, so there has been much research on minimizing energy usage by introducing sleep times, reducing the amount of transmitted data, etc. While those topics remain important, other aspects have been neglected. With long sleep times come long delays, which can grow rapidly with the size of the network, depending on its topology.

Traffic flows as well as scheduling regimes have so far often been considered fluid, neglecting aspects like packetizing or TDMA and describing them only by their sustained maximum or average rate and modifiers like a burst for incoming traffic or latencies for services, which is a good approximation for fast and relatively fine-grained data streams. When looking at very slow data streams however, that model loses precision. With slow

medium data rates and processing, data sent over such a network loses its fluid characteristics. This becomes a concern in TDMA systems, where participants only get intermittent service during their assigned time slots, and need to be quiet outside those slots. Modeling such a system with fluid models fails to take this into consideration.

In this paper, we first give an overview of the necessary methodological background for worst-case analysis of WSNs in section 2. In section 3, we present the optimization problem of finding the most energy-preserving frame length in a TDMA system while still meeting worst-case delay constraints. In Section 4, this problem is solved analytically to find solutions for a general sink tree network under the fluid model. Finally, in section 5, we present an implementation using the existing DISCO Network Calculator framework [1], allowing us to analyze the impact of discrete models on worst-case delay bounds. This provides us with a means to compute bounds for TDMA parameters in sink-tree sensor networks. A few numerical examples analyzing different parameters of the WSN are also discussed in this section.

The problem we address in this paper is different from other approaches that try to find optimally fast solutions, like e.g. [2], in that we find the *minimum* medium allocation that still allows the WSN to work within the given limits. Other related work covers aspects of TDMA networks, like time synchronization [3] or resource allocation and reuse in spatially spread out networks [4,5]. The general tenor is to maximize performance of a network, whereas we aim for the minimum performance at which quality of service requirements, in this case delay, are stil fulfilled.

2 Sensor Network Calculus

In this section, we assume basic familiarity with the concepts of network calculus as described in [6]. A quick rundown of the most important aspects is given in Appendix A. For sensor networks, the calculus has been extended by some aspects which are described in the following paragraphs.

2.1 Sensor Network System Model

In this paper the common class of single base station oriented operation models is assumed, with networks being structured as shown in Figure 1. Within the traffic that is modeled only the sensor reports are taken into account. Traffic generated from the base station towards the nodes (e.g. topology control and node configuration) can be taken into account by

factoring its effect into the service curves of the nodes by e.g. assuming strict priority for that kind of traffic. Furthermore, it is assumed that the routing protocol being used forms a tree in the sensor network. Hence n sensor nodes arranged in a directed acyclic graph are given, as shown in Figure 1.

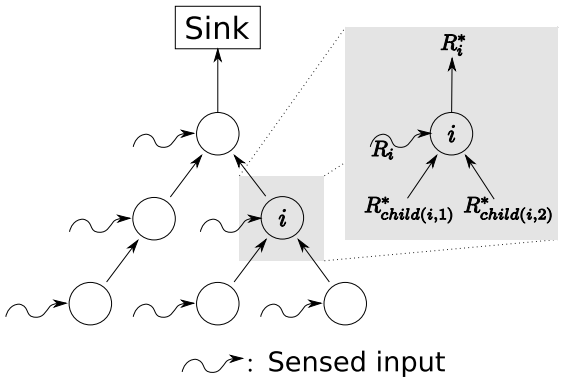

Fig. 1. Sensor network model

Each sensor node i senses its environment and thus is exposed to an input function R_i corresponding to its sensed input traffic. If sensor node i is not a leaf node of the tree then it also receives sensed data from all of its child nodes $child(i,1),\ldots,child(i,n_i)$, where n_i is the number of child nodes of sensor node i. Sensor node i forwards/processes its input which results in an output function R_i^* from node i towards its parent node.

Now the basic network calculus components, arrival and service curve, have to be incorporated. First the arrival curve $\bar{\alpha}_i$ of each sensor node in the field has to be derived. The input of each sensor node in the field, taking into account its sensed input and its children's input, is given by:

$$\bar{R}_i = R_i + \sum_{j=1}^{n_i} R^*_{child(i,j)} \tag{1}$$

Thus, the arrival curve for the total input function for sensor node i is given by:

$$\bar{\alpha}_i = \alpha_i + \sum_{j=1}^{n_i} \alpha^*_{child(i,j)} \tag{2}$$

For this paper, we use simple token-bucket functions to model inputs. Those are defined over $\mathbb{R}_0^+$as

$$\gamma_{r,b}(t) = \begin{cases} 0 & \text{if } t = 0 \\ b + rt & \text{otherwise} \end{cases}$$

Second, the service curve has to be specified. The service curve depends on the way packets are scheduled in a sensor node which mainly depends on link layer characteristics. More specifically, the service curve depends on how the duty cycle and therefore the energy-efficiency goals are set.[1] For this paper, we will use a service curve modeling the periodic availability of the full medium capacity C after an initial delay T – this closely captures the TDMA characteristics we assume. The form of this curve for a node receiving s time units of service in a frame of duration f is shown in Figure 2. Until point T (which will usually be $T = f - s$ to express the maximum access latency in a TDMA network), the curve is at zero. From T to $t + s$, it rises to sC and then remains there until $T + f$, where it starts to rise again. This repeats forever with a period length of f. The structure is similar to the one proposed in [7] for ZigBee networks. This curve can be approximated by a rate-latency curve $\beta_{R,T}(t) = max\{R(t-T), 0\}$ with $R = \frac{s}{f}C$ and $T = f - s$. That curve is shown labeled as β in the figure and can be considered the fluid version of the TDMA service curve.

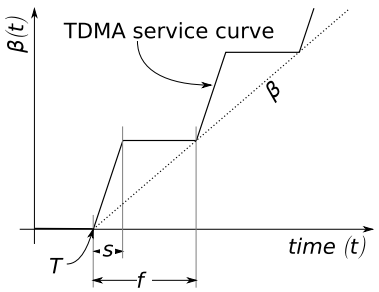

Fig. 2. TDMA service curve.

Finally, the output of sensor node i, i.e. the traffic which it forwards to its parent in the tree, is constrained by the following arrival curve:

[1] The service curve might further depend on whether more advanced sensor network characteristics like in-network processing, e.g. for aggregation or even prioritization of some traffic is provided.

$$\alpha_i^* = \bar{\alpha}_i \oslash \beta_i = \left(\alpha_i + \sum_{j=1}^{n_i} \alpha^*_{child(i,j)} \right) \oslash \beta_i \tag{3}$$

In order to calculate a network-wide characteristic like the maximum information transfer delay or local buffer requirements especially at the most challenged sensor node just below the sink (which is called node 1 from now on) the iterative procedure in Algorithm 1 is used to calculate the network internal flows.

Algorithm 1 Calculating the internal flows of a network

1. Let us assume that arrival curves for the sensed input α_i and service curves β_i for sensor node i, $i = 1, \ldots, n$, are given.
2. For all leaf nodes the output bound α_i^* can be calculated according to (12). Each leaf node is now marked as "calculated".
3. For all nodes only having children which are marked "calculated" the output bound α_i^* can be calculated according to (3) and they can again be marked "calculated".
4. If node 1 is marked "calculated" the algorithm terminates, otherwise go to step 3.

After the network internal flows are computed according to this procedure, the local worst case buffer requirements B_i and per node delay bounds D_i for each sensor node i can be calculated according to Theorem 1 and 2:

$$B_i = v(\bar{\alpha}_i, \beta_i) = \sup_{s \geq 0} \{ \bar{\alpha}_i(s) - \beta_i(s) \} \tag{4}$$

$$D_i = h(\bar{\alpha}_i, \beta_i) = \sup_{s \geq 0} \{ \inf \{ \tau \geq 0 : \bar{\alpha}_i(s) \leq \beta_i(s + \tau) \} \} \tag{5}$$

To compute an end-to end delay in a network where data is forwarded unchanged, we can use the result for the so-called Pay Multiplexing Only Once analysis (PMOO) described in [8]. Due to the simple structure of the network, all flows that join a flow of interest will remain multiplexed until the sink, making it possible to simplify the analysis to Algorithm 2. When compared to the addition of the nodal delay bounds from (5) or the direct application of the blind multiplexing result from Theorem 5 at each of the nodes this results in considerably less pessimistic bounds, because each interfering flow's burst has to be taken into consideration only once (for an in-depth comparison see [8]).

Algorithm 2 Simplified PMOO Analysis.

1. Let $M = \{E_1, \ldots, E_n\}$ be the set of edges the flow of interest is traversing on the way from its source to the sink. Each edge E_i has an incoming node N_{i-1} and an outgoing node N_i.
2. Let $\beta^0_{\text{eff}} = \delta_0$with
$$\delta_d(t) = \begin{cases} 0 & \text{if } t \leq d \\ \infty & \text{otherwise} \end{cases}$$
δ_0 is the neutral element of the min-plus convolution.
3. For all $E_{1 \leq i \leq n} \in M$, add up all upper output bounds from incoming nodes $N \neq N_{i-1}$ (for $i = 0$ this means the sum of all incoming flows except the flow of interest) and update the effective service curve:
$$\beta^i_{\text{eff}} = \left[\left(\beta^{i-1}_{\text{eff}} \otimes \beta_{E_i} \right) - \sum_{N \neq N_{i-1}} \alpha^*_N \right]^+$$
with the α^*_N according to equation 3.
4. $\beta_{\text{eff}} = \beta^n_{\text{eff}}$ is the effective service curve for the flow of interest.

3 Optimal TDMA Design

When designing a TDMA system, a choice has to be made for how long the repetitive TDMA frame as well as the individual slot sizes of each participating node are. Since the advantage of TDMA systems against concurrent medium access lies in the fact that each participant obtains exclusive use of the medium, it has to be ensured that each participant gets assigned enough time to perform its tasks. For some network nodes, that just requires a short slot in which they can send collected data, and perhaps receive an acknowledgment from an upstream node. However, in multi-hop systems, some nodes act as routers, and have higher bandwidth requirements for forwarding other nodes' data, while perhaps collecting and sending data themselves. Aside from avoiding contention, using TDMA also reduces energy consumption by making it possible for nodes to power down in periods without relevant traffic.

Since in wireless sensor networks, two main concerns are minimizing power consumption and meeting delay bounds, while transmission bandwidth requirements tend to be low, we want to maximize the frame length, giving the sensor nodes the opportunity to disable their radio transceivers or even go into deep sleep modes.

3.1 General TDMA Design Problem

From those requirements, we formulate the TDMA design problem as an optimization problem for a tree network with n nodes where from each node a flow is originating:

$$\begin{array}{lll} \text{max. } Z = \min_{1\leq i\leq n}\{f - s_i\} & & \\ \text{s.t.} \quad \sum_{i=1}^{n} s_i \leq f & & \text{(TDMA integrity)} \\ \quad \forall i: \ d_i(f, \boldsymbol{s}|r, b, C) \leq D & & \text{(Delay)} \\ \quad \forall i: \ \frac{s_i}{f} \cdot C \geq F_i r & & \text{(Rate)} \\ \quad \forall i: s_i \geq 0, \ f \geq 0 & & \text{(Non-negativity)} \end{array}$$

In all further instantiations of this problem, the constraints will be given in the same order. The designations have been omitted due to typesetting issues. Here, f is the length of the repetitive TDMA frame, s_i is the amount of time devoted to node i for sending (slot size of node i). These constitute the decision variables. For the parameters of the problem we further have, D as the maximum permissible delay that may be incurred by any flow in the sensor field, $d_i(f, \boldsymbol{s}|r, b, C) = h(\gamma_{r,b}, \beta_{\text{eff},i})$ as the actual maximum delay incurred by flow i (which is computed based on the effective service algorithm described in the previous section, Algorithm 2), F_i as the number flows carried by node i (including the flow originating at node i), C as the medium rate ("capacity"), and r as the (maximum) sustained rate for any flow as well as b as the maximum burst of a flow. Note that we assume the sensors to have identical arrival curves $\gamma_{r,b}$ as well as an identical delay requirement D, which for most practical situation will be no restriction and makes the further analysis more tractable.

The objective function reflects the fact that the minimum sleeping period over all nodes in the field should be maximized, thus achieving a maximum lifetime of the network. The TDMA integrity constraint captures the fact that all slot sizes together must fit into the TDMA frame. Obviously the delay constraints should be met for all flows, which is captured by the delay constraints, as well as all the rate constraints must be met in order not to obtain infinite delay bounds for the flows. Of course, we also have non-negativity constraints for the decision variables. As a remark, in the objective function there is a hidden assumption on the relative energy costs for switching between different states like transmission, reception, and idle. It is assumed that those state transitions have roughly the same cost, as it is the case for many transceiver architectures (see e.g. [9]), such that by minimizing the sleep period of a node with respect to sending will in fact coincide with minimizing the amount of energy con-

sumed for transmission *and* reception by maximally batching data before forwarding them (within the delay constraints).

Unfortunately, this general modeling of the TDMA design problem results in a very hard to solve non-linear programming problem with $n+1$ decision variables and $3n+2$ constraints. The non-linearity is exhibited in the objective function as well as in the delay constraints. Hence, the only viable approach is to simplify the problem structure if a solution shall be found for larger instances of the TDMA design problem. There are two intuitive approaches towards relaxing the problem:

1. *Equal Slot Sizing (ESS):* the assignment may be made such that inside a fixed time slot length, each node can transmit enough data to fulfill all requirements.
2. *Traffic-Proportional Slot Sizing* (TPSS): slots may be assigned such that each node only claims the resources necessary to fulfill its own duties, depending on the input bandwidth and forwarded data streams.

While the second relaxation approach may appear more efficient, it is also harder to set up. The first approach requires rather little information – the number of nodes and the bandwidth requirements of the node serving the highest number of flows –, the second method requires good knowledge of the topology, which may not always be at hand.

In both cases two variables need to be controlled: The overall frame length f and the individual slot length s, where however for ESS this slot length is for each node, while for TPSS each node obtains a multiple of that slot length depending on the number of flows it has to carry. Obviously in both cases, with an increasing frame length, a node may sleep longer between transmission or reception phases, but delay is increased at the same time. For a given f in a network with n nodes, s is limited to values between an upper bound $\frac{f}{n}$ and a lower bound that is given by the minimum bandwidth requirements.

Next we state the TDMA design problem under ESS as well as TPSS, which as we will see in Section 4 are amenable to an analytical solution under TDMA service curves approximated by rate-latency service curves.

3.2 TDMA Design under Equal Slot Sizing

Under ESS, we now consider a common slot size s for all nodes. The TDMA design problem can then be formulated as:

$$\begin{aligned}
\text{max. } & Z = f - s \\
\text{s.t. } & s \leq \tfrac{f}{n} \\
& d(f, s|r, b, C) = \max_{1 \leq i \leq n} d_i(f, s|r, b, C) \leq D \\
& \tfrac{s}{f} \cdot C \geq F_{max} r \\
& f \geq 0
\end{aligned}$$

We obtain an optimization problem with a linear objective function, and only four constraints – a great simplification. Most of the reduction in complexity is due to lower number of decision variables as e.g. for the rate constraints which collapse into a single one. For the delay constraints this is not as easily seen but the reader may ascertain herself that there is always one flow which is the worst-case flow and whose delay constraint is dominating all of the other flows' delay constraints as they are all facing the same situation when considering the parameters. For example in a fully occupied n-ary tree one can choose any leaf node and its corresponding flow as the flow whose delay constraint is the dominating one.

3.3 TDMA Design under Traffic-Proportional Slot Sizing

Under traffic-proportional slot sizing, each flow is allocated a slot of size s on each node that it passes on its way to the sink, i.e. we obtain the following relaxation of the TDMA design problem:

$$\begin{aligned}
\text{max. } & Z = f - s \cdot \max_{1 \leq i \leq n} \{F_i\} \\
\text{s.t. } & \textstyle\sum_{i=1}^{n} F_i s \leq f \\
& d(f, s|r, b, C) = \max_{1 \leq i \leq n} d_i(f, s|r, b, C) \leq D \\
& \tfrac{s}{f} \cdot C \geq r \\
& f \geq 0
\end{aligned}$$

Again, we obtain a problem with a linear objective function, and only four constraints, which very much exhibits the same shape as the TDMA design problem under ESS following similar arguments. In particular, the dominating delay constraint must again be found, which often becomes an obvious task in regular topologies.

4 Analytical Solution in Fluid Setting

In the following we first discuss the relative merits of the equal and traffic-proportional slot sizing relaxations in a simple, yet illustrative example of a two-hop sensor network as shown in Fig. 3. We assume the rate-latency

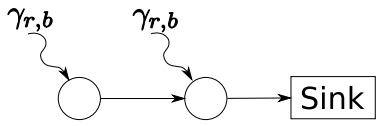

Fig. 3. Two node example network

curves as fluid approximations for the TDMA service in order to keep the problem analytically tractable. Since we will find that the equal slot sizing results in more energy-efficient schedules, we then show how in the general sink tree case of fluid service curves we can derive an optimal solution. Based on the insight from this analytical solution for rate-latency service curves we will then develop a numerical solution for the more involved case of accurately modeling service curves resulting from the time-slotted nature of the MAC.

4.1 Equal vs. Traffic-Proportional Slot Sizing

Since both equal and traffic-proportional slot sizing have their intuitive rationale we compare them to each other in a simple two-hop network as shown in Fig. 3. For both relaxations we derive their analytical optimal solution for this example scenario. Apart from the comparison of the two we will also gain insights on how to solve the respective optimization problems analytically in the general case.

Equal Slot Sizing Under equal slot sizing and in the two-hop network we obtain the following incarnation of the optimal TDMA problem:

$$\begin{aligned} \max.\ & Z = f - s \\ \text{s.t.}\ \ & s \leq \tfrac{f}{2} \\ & d_2(f,s|r,b,C) = \tfrac{\frac{s}{f}C(f-s)+2b}{\frac{s}{f}C-r} + f - s \leq D \\ & \tfrac{s}{f} \cdot C \geq 2r \\ & f \geq 0 \end{aligned}$$

Since the objective function is linear the solution to this optimization problem must lie on the border of the feasible region (it is guaranteed to exist since the feasible region is closed). In figure 4 the feasible region as well as a contour line of the objective function are drawn (for $r = 1, b = 1, C = 10,\ D = 1$). It can be seen that the optimum must be

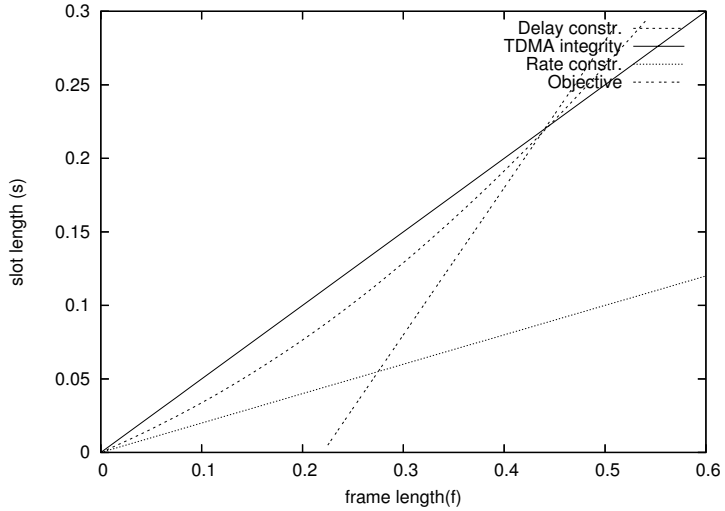

Fig. 4. Graphical illustration of the optimization problem for ESS. The feasible region is above the delay constraint and below the TDMA integrity constraint. The rate constraint does not affect the feasible region in this example.

taken on at the lower border of the feasible region. In fact, we can write the border constituted by the delay constraint as shown in equation 6 because the delay constraint is a quadratic form in s and f which can be solved for s with two real solutions of which we take the larger one as it results in a more binding constraint. A moment's consideration exhibits that $\forall f : \frac{\partial g}{\partial f} < 1$ since otherwise an increase in frame size would result in a larger increase of the slot size, which obviously cannot be the case. On the other hand the partial derivative of the objective function after f is 1, which means that the optimum must be taken on at the corner point of the feasible region where delay constraint and TDMA integrity constraint intersect. In other words, the *TDMA design problem under ESS* can be reduced to *matching the delay with the TDMA integrity constraint.*

$$\begin{aligned} s &= g(f|r,b,C,D) \\ &= \frac{\sqrt{C^2D^2 + (6frC - 4fC^2)D + 4f^2C^2 + (16bf - 4f^2r)C + f^2r^2} + 2fC - CD + fr}{4C} \end{aligned} \tag{6}$$

Traffic-Proportional Slot Sizing Under traffic-proportional slot sizing and in the two-hop network, we obtain the following incarnation of the optimal TDMA problem:

$$\begin{aligned} \text{max. } & Z = f - 2s \\ \text{s.t. } & s \leq \tfrac{f}{3} \\ & d_2(f,s) = \tfrac{b}{\frac{s}{f}C} + \tfrac{2\frac{s}{f}C(f-2s)+b}{2\frac{s}{f}C-r} + f - s \leq D \\ & \tfrac{s}{f} \cdot C \geq r \\ & f \geq 0 \end{aligned}$$

Again, as the objective function is linear the solution to this problems must be taken on at the border of the feasible region. In figure 5, the feasible region as well as some contour lines of the objective function are displayed. Note that we exchanged the axes, as for this case it was hard

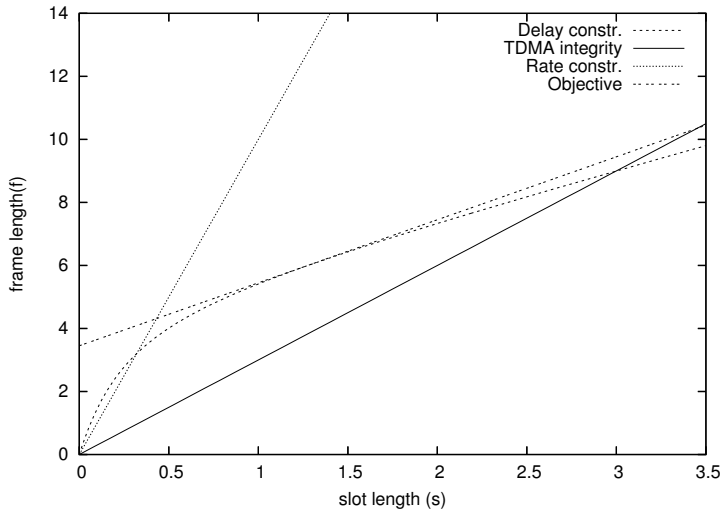

Fig. 5. Graphical illustration of the optimization problem for TPSS. The feasible region is contained in the area above the TDMA constraint and below delay and rate constraint.

to find a closed form of the delay constrained solved after s as it resulted

in imaginary solutions, whereas a resolution after f was possible again. That means here we need to search for the optimum at the upper border of the feasible region. Furthermore, here we cannot argue that the optimum solution has to be on a cornerpoint of the feasible region but may be taken on at $f = g(s|r, b, C, D)$, the border constituted by the delay constraint. Hence, first we have to compute the maximum for $g(s|r, b, C, D) - 2s$, which is a straightforward one-dimensional optimization problem. Then it must be checked if the solution is feasible, otherwise rate or TDMA constraint have to be matched with the delay constraint (see again Fig. 5 and imagine e.g. a tighter rate constraint).

Numerical Comparison of ESS and TPSS In order to compare the ESS and TPSS relaxations Table 1 provides some numerical examples since a general investigation is hard to conduct.

				TPSS			ESS		
C	r	b	D	f	s	Z	f	s	Z
10	1	1	1	0.48	0.16	0.16	0.44	0.22	0.22
10	1	1	5	2.99	0.7	1.59	4	2	2
10	1	1	10	6	1.28	3.44	8.44	4.22	4.22
10	1	1	20	12.05	2.43	7.19	17.33	8.67	8.66

Table 1. Analytical results

Interestingly, despite the more intuitively more appealing nature of the TPSS relaxation, the ESS relaxation achieves better results with respect to maximizing the minimum sleep period in the field. As it is also the easier one to solve we further on focus on the ESS relaxation for solving the TDMA design problem.

4.2 Analytical Solution for ESS in General Sink Trees

What remains to do in general sink trees compared to the two-hop network in the previous setting is to show that the delay constraint again takes on a quadratic form. This then allows to easily express the slot size as a function of the frame size and the same arguments as in the two hop case will lead to the conclusion that the optimum solution is given at the point where delay and TDMA integrity constraint are matched. Hence let us discuss the delay constraint in a general sink tree network:

We assume a general sink tree network with each node offering a service curve $\beta_{\frac{s}{f}C,f-s}$ and flows starting from each node constrained by arrival

curve $\gamma_{r,b}$. Looking at a particular flow we have a situation as depicted in Fig. 6. Applying the PMOO analysis results in the following effective

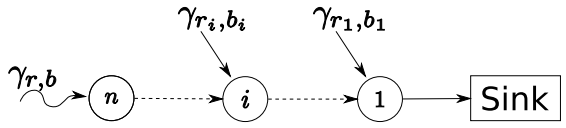

Fig. 6. Flow in a general sink tree

service curve for the flow of interest:

$$\begin{aligned}\beta_{\text{eff},n} &= \left[\left[\beta_{R,T} - \gamma_{r_1,b_1}\right]^+ \otimes \beta_{R,T} - \gamma_{r_2,b_2}\right]^+ \otimes \dots \\ &= \beta_{R-\sum_{i=1}^{n-1} r_i,\, \frac{T\left(nR-\sum_{i=0}^{n-1}\sum_{j=1}^{i} r_j\right)+\sum_{i=1}^{n-1} b_i}{R-\sum_{i=1}^{n-1} r_i}}\end{aligned}$$

with $R = \frac{s}{f}C$ and $T = f - s$ and $r_i = a_i r$ and $b_i = c_i b + d_i r T$. For the latter expressions the parameters $a_i, c_i, d_i \in \mathcal{N}$ are depending on the topology. The delay constraint for flow n (i.e the one originating at node n) can thus be expressed as shown in equation 7.

While seemingly a complex expression this constitutes again a quadratic form in f and s which can be recast as shown in equation 8 where we can ignore the smaller right hand side version as the larger one is the physically meaningful. Hence, we can make the same observations as in the 2-hop network case and again reason that the optimal solution must be taken on at the cornerpoint of the feasible region where TDMA integrity and delay constraint are matched.

5 Numerical Approach in the Discrete Setting

For the numerical approach, we extended the DISCO Network Calculator [1] to handle affine curves as described in [10], which allows us to model discrete service and arrival curves and perform network analysis with such. We specifically model the service curves as shown in Figure 2. The arrival curve is modeled as a simple token bucket, mostly to reduce computing time. It also resembles the behavior of a sensor that is constantly doing measurements, like a low-bandwidth audio stream or some other task that requires frequent sampling.

The TDMA design problem, based on the insight from the previous section that TDMA integrity and delay constraint have to be matched,

$$
\begin{aligned}
h\left(\gamma_{r,b},\beta_{\text{eff},n}\right) &= \frac{b}{\frac{s}{f}C - r\sum_{i=1}^{n-1} a_i} \\
&\quad + \frac{(f-s)\left(n\frac{s}{f}C - r\sum_{i=0}^{n-1}\sum_{j=1}^{i} a_j\right) + b\sum_{i=1}^{n-1} c_i + r(f-s)\sum_{i=1}^{n-1} d_i}{\frac{s}{f}C - r\sum_{i=1}^{n-1} a_i} \\
&\leq D
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
s &\geq \frac{\sum d_i - CD + fnC + fr\sum a_i - fr}{2nC} \\
&\quad \pm \frac{\sqrt{f^2r^2\left(\sum d_j\right)^2 + A_1 + C^2D^2 + A_2 + (f^2n^2 + 4bfn)C^2 + A_3 + f^2r^2(\sum\sum a_j)^2}}{2nC}
\end{aligned} \tag{8}
$$

with

$$
\begin{aligned}
A_1 &= (2frCD + 2f^2nrC - 2f^2r^2\sum\sum a_j)\sum d_i \\
A_2 &= ((4fnr\sum a_i - 2fr\sum\sum a_j)C - 2fnC^2)D \\
A_3 &= (4bfn - 2f^2nr\sum\sum a_j)C
\end{aligned}
$$

ultimately boils down to a root-finding problem: a black-box function $d(f, s|r, b, C, D)$ returns the delay incurred for the parameters. Since we have seen that the optimum is found at $s = \frac{f}{n}$, this is a function with one variable $d(f|r, b, C, D)$. To maximize the frame length f, we seek a solution $f'$for which $d(f'|.) = D$, which is a root for $d(f|.) - D = 0$. We further know that $\forall 0 < f < f' :\ d(f|.) - D < 0$ and $\forall f > f' :\ d(f|.) - D > 0$. Thus, we can use interval bisection to find the root. We are specifically using an algorithm that approaches the root from below, making sure that the calculated value is smaller than f', ensuring it is inside the feasible region.

5.1 Ultimately Affine Piecewise Linear Curves

[10] describes algorithms for convolution and deconvolution of (ultimately) pseudo-periodic functions. It also mentions *ultimately affine* functions that end with an infinite affine segment, but that class of functions is not explicitly handled in the algorithms. We will now show that the theorems pertaining to convolution and deconvolution also hold when infinite segments are involved.

Lemma 1. *Convolution of a spot and an infinite segment. Let f_1 be a segment on $]a, +\infty[$ and f_2 be a sppot on c. Then, $f_1 \otimes f_2$ is a segment on $]a + c, +\infty[$ abd $\forall t \in]a, +\infty[:\ f_1 \otimes f_2(c + t) = f_1(t) + f_2(c)$.*

Proof. The proof is the same as in the finite case.

Lemma 2. *Convolution of segments. Let* $f_1,\ f_2 \in \mathcal{F}$ *be two segments on respectively* $]a,b[$ *and* $]c,d[$ *with* $a,c \in \mathbb{R}_0^+$ *and* $b,d \in \mathbb{R}_0^+ \cap \{\infty\}$ *(meaning that either segment may be infinite), and slopes* r_1 *and* r_2*, so that* $r_1 \leq r_2$*. Then* $f_1 \otimes f_2$ *is equal to* $+\infty$ *outside* $]a+c, b+d[$*, while* $\forall t \in]a+c, b+d[$:

$$f_1 \otimes f_2(t) = \begin{cases} f_1(a+) + f_2(c+) + r_1(t-a-c) & \text{if } t \leq b+c \\ f_1(a+) + f_2(c+) + r_1(b-a) + r_2(t-b-c) & \text{if } t > b+c \end{cases} \tag{9}$$

Proof. While this is the same equation as in [10], we need to describe the behaviour for the case that at least one segment is infinitely long. By definition, $f_1 \otimes f_2(t) = \inf_{0 \leq s \leq t}\{f_1(s) + f_2(t-s)\}$. It still holds that $\forall t \leq a+c:\ f_1 \otimes f_2(t) = \infty$. If both segments are finite, i.e. both b and d are finite, the situation is obviosly exactly the same as in the original theorem. If at least one of the segments is infinite (so $\max\{b,d\} = \infty$), the support of $f_1 \otimes f_2$ is $]a+c, \infty[$. There are now two interesting cases to look at: b is finite and d is infinite, or b is infinite (as we will see, d does not matter in that situation).

In case b is infinite, the infimum in the convolution is always achieved at $t-s=c$ because $r_1 \leq r_2$. Thus, $\forall t \in]a+c, \infty[:\ f_1 \otimes f_2(t) = f_1(t-c) + f_2(c+)$, proving eq. 9 for that case.

If b is finite, the inflection point at $b+c$ becomes important. The proof for this case is analog to the original proof, since that is independent of the upper limit of the support $(b+d)$. □

In both cases, the geometric interpretation remains: starting from $a+c$ on the time axis and $f_1(a+) + f_2(c+)$ on the y-axis, the segments are concatenated in order of increasing slope. If the first segment is infinite, the second becomes irrelevant.

The theorems for the deconvolution can be shown to properly handle infinite segments in a similar manner. The geometric interpretation remains the same too, except that the inflection point must be used as the point of origin for infinite segments when deconvoluting two segments. A slightly interesting case occurs for two infinite segments f_1 and f_2 with slopes $r_1 > r_2$. This results in $\forall t \in \mathbb{R}_0^+:\ f_1 \oslash f_2(t) = \infty$.

5.2 Numerical Examples

Using that procedure, we set up a number of sample networks, among them the one from the previous section and fully populated binary trees

of depth 3 and 5^2 and analyzed those in the enhanced DISCO Network Calculator. The results for those test runs can be found in Table 2. The

Setting	C	r	b	D	fluid f	discrete f
2 nodes	10	1	1	1	0.4444	0.7368
$\frac{f}{s} = 2$	10	1	1	5	4.0	4.5263
Binary tree, 3 deep	5000	1	1	10	3.5356	3.5859
$\frac{f}{s} = 14$	5000	1	1	50	17.7062	17.9315
Binary tree, 5 deep	5000	1	1	10	1.2811	1.4435
$\frac{f}{s} = 62$	5000	1	1	50	6.7394	7.2209

Table 2. Comparison of numerical results in fluid and discrete settings

number $\frac{f}{s}$ given for each scenario denotes the number of slots a frame is split into; values for s are not explicitly given. Obviously, larger values for f mean larger values for the objective function too. From those numbers, it becomes apparent that a discrete analysis has an advantage over an analysis in the fluid domain. The magnitude of that advantage depends on the scenario.

Figure 7 explains that improvement. As seen in section 2, the latency

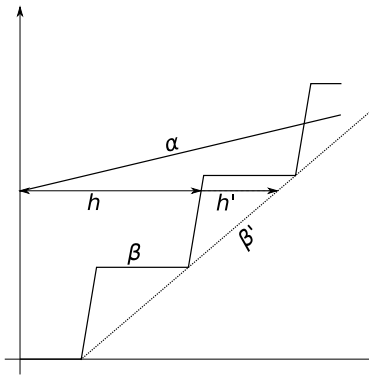

Fig. 7. Illustration of the improvement

bound is computed as the maximum horizontal distance between the arrival curve α and the service curve. In the fluid setting, the service curve

[2] The binary trees are slightly irregular as the root node doesn't have an input and acts as the sink. This explains the number of slots $\frac{f}{s}$ shown in the table, which for a tree depth of d is $2^{d+1} - 2$.

models the average medium rate $\frac{s}{f}C$ available to a node, as shown by the curve β'; in the discrete setting, a node periodically gets the full rate C, which makes the discrete service curve β "jump" above β'. Because of this, it holds that $\forall \alpha \in \mathcal{F}: \; h(\alpha, \beta) \leq h(\alpha, \beta')$.

It is apparent from the numerical results that for larger networks, the relative advantage of a discrete analysis becomes smaller. The same holds for service curves with a lower step height. This is because, due to the nature of the effective end-to-end service curve, the step size remains the same, but the latency grows with each hop and interfering flow. When referring to Figure 7, h grows faster than $h' - h$.

It is also notable that in our test-cases, the optimal value for f scales linearly with D, so for a given network with all other parameters fixed, the frame length can be easily extrapolated from a few computed values. The behavior with changing C is potentially more interesting, since a modified service curve is accompanied with it. However, when looking at a sample data set in Figure 8, which was created by varying C from 1000 to 10000 in steps of 100 with $r = 1$, $b = 1$ and $D = 10$, one notices that the gains in the possible frame lengths are minimal and even diminishing for larger rates. Such small gains in frame length are likely outweighed by the rising energy requirements imposed by the hardware necessary to achieve a faster transmission speed.

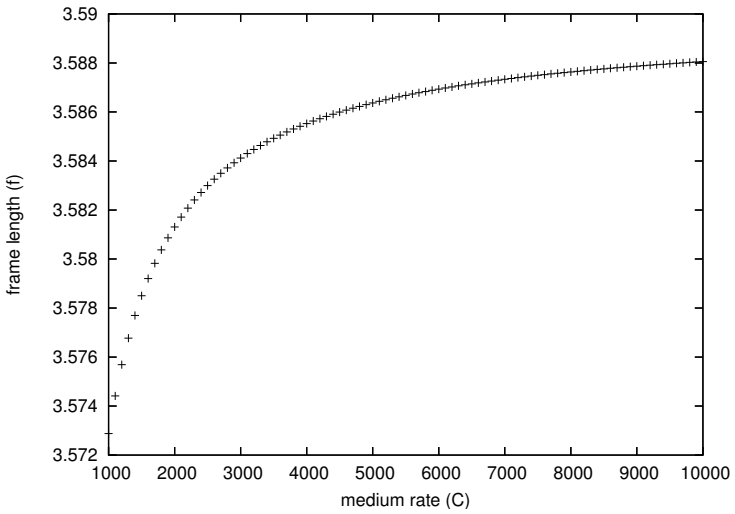

Fig. 8. Frame lengths for varying medium speed

Of course the arrival curve α could be modeled discretely too, in a similar way as the service curve, below the fluid arrival curve α. It is true that this would refine the results further, but due to the nature of the calculations involved, this approach tends to be computationally extremely expensive. For addition or subtraction of two curves with periodic parts of duration d_1 and d_2, the resulting curve will have a period duration of $\mathrm{scm}(d_1, d_2)$, with scm being the smallest common multiple. Due to the potentially large number of calculations involved in a network analysis, we observed durations of 10^{15} time units and more, with a lot of short segments, which was very taxing on computational resources without producing any usable results.

6 Conclusions

Providing real-time guarantees in WSNs while still optimizing energy-efficiency has turned out to be a non-trivial problem in TDMA-based WSNs. Nevertheless, based on the general problem statement in the form of a non-linear mathematical program, we were able to derive an optimal solution for a relaxed version of the problem that achieves very good energy-efficiency, i.e. long sleeping periods for the sensor nodes, while still meeting the real-time constraints. We provided an analytical solution for the TDMA design optimization problem that handles the calculation for generic sink-tree WSNs with fluid models. Based on this, we have created a means to find numerical solutions for the TDMA design problem with discrete TDMA service curves by extending an existing framework, allowing to perform the computation for arbitrary sink-tree WSNs.

Our results show that using a discrete traffic model leads to less restrictive parameters, thus making longer inactivity periods possible and allowing sensor nodes to sleep, prolonging their lifetime.

Future Work

During the implementation of affine curve algorithms in the DISCO Network Calculator, a few potential shortcomings have emerged. Most of all, the algorithms tend to have a high time complexity due to a quick growth of period descriptions. In this area, further work is necessary to find better representations and algorithms for special classes of curves, like the one presented in this paper. On the analytical side, it has become apparent that even simple networks may give rise to very hard optimization problems. Further simplifying those problems within tolerable bounds may

yield computationally less expensive problems for some types of sensor networks.

It is also worthwhile to look into scheduling algorithms, as an integrated part of the problem or as an addition, to minimize state transitions on the network nodes. As mentioned, we make the assumption that changing between transmission, reception and sleep states has the same cost in terms of energy, so ordering the nodes optimally is less of a concern. This order inside a TDMA frame can also be optimized to speed up packet transfer through a network [11]. Integrating such techniques into our framework could lead to improved bounds.

References

1. J. B. Schmitt and F. A. Zdarsky, "The DISCO Network Calculator - A Toolbox for Worst Case Analysis," in *Proceedings of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06), Pisa, Italy*, ACM, Nov. 2006.
2. S. Cui, A. Madan, R.and Goldsmith, and S. Lall, "Energy-delay tradeoffs for data collection in TDMA-based sensor networks," in *ICC 2005 Vol. 5*, IEEE, May 2005.
3. M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," 2003.
4. P. Björklund, P. Värbrand, and D. Yuan", "Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach."
5. S. Ergen and P. Varaiya, "TDMA scheduling algorithms for sensor networks," technical report, University of California, Berkley, 2005.
6. J.-Y. L. Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. New York, NY, USA: Springer-Verlag New York, Inc., 2001.
7. *Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks*, (Washington, DC, USA), IEEE Computer Society, 2006.
8. J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Performance Bounds in Feed-Forward Networks under Blind Multiplexing," Technical Report 349/06, University of Kaiserslautern, Germany, Apr. 2006.
9. A. Wang and C. Sodini, "A simple energy model for wireless microsensor transceivers," in *Global Telecommunications Conference 2004*, IEEE, IEEE, November 2004.
10. A. Bouillard and E. Thierry, "An algorithmic toolbox for network calculus," tech. rep., Unité de recherche INRIA Rennes, 2007.
11. A. Giusti, A. Murphy, and G. Picco, "Decentralized Scattering of Wake-up Times in Wireless Sensor Networks," in *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN), Delft, The Netherlands, Lecture Notes on Computer Science, Volume 4373*, pp. 245–260, Springer, Jan. 2007.

A Network Calculus

Network calculus is *the* tool to analyze flow control problems in networks with particular focus on determination of bounds on worst case

performance. It has been successfully applied as a framework to derive deterministic guarantees on throughput, delay, and to ensure zero loss in packet-switched networks. Network calculus can also be interpreted as a system theory for *deterministic* queuing systems, based on min-plus algebra. What makes it different from traditional queuing theory is that it is concerned with worst case rather than average case or equilibrium behavior. It thus deals with bounding processes called arrival and service curves rather than arrival and departure processes themselves.

Next some basic definitions and notations are provided before some basic results from network calculus are summarized.

Definition 1. *Class of functions $\mathcal{F}$. The following operations are defined over the class of functions in:*

$$\mathcal{F} = \left\{ f : \mathbb{R}^+ \to \mathbb{R}^+,\ \forall t \geq s :\ f(t) \geq f(s),\ f(0) = 0 \right\}$$

The input function $R(t)$ of an arrival process is the number of bits that arrive in the interval $[0, t]$. In particular $R(0) = 0$, and R is wide-sense increasing, i.e., $R(t_1) \leq R(t_2)$ for all $t_1 \leq t_2$.

Definition 2. *The output function $R^*(t)$ of a system S is the number of bits that have left S in the interval $[0, t]$. In particular $R^*(0) = 0$, and R is wide-sense increasing, i.e., $R^*(t_1) \leq R^*(t_2)$ for all $t_1 \leq t_2$.*

Definition 3. *Min-Plus Convolution. Let f and g be wide-sense increasing and $f(0) = g(0) = 0$. Then their convolution under min-plus algebra is defined as*

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\}$$

Definition 4. *Min-Plus Deconvolution. Let f and g be wide-sense increasing and $f(0) = g(0) = 0$. Then their deconvolution under min-plus algebra is defined as*

$$(f \oslash g)(t) = \sup_{s \geq 0} \{f(t+s) - g(s)\}$$

Now, by means of the min-plus convolution, the arrival and service curve are defined.

Definition 5. *Arrival Curve. Let α be a wide-sense increasing function such that $\alpha(t) = 0$ for $t < 0$. α is an arrival curve for an input function R iff $R \leq R \otimes \alpha$. It is also said that R is α-smooth or R is constrained by α.*

Definition 6. *Service Curve. Consider a system S and a flow through S with R and R^*. S offers a service curve β to the flow iff β is wide-sense increasing and $R^* \geq R \otimes \beta$.*

From these, it is now possible to capture the major worst-case properties for data flows: maximum delay and maximum backlog. These are stated in the following theorems.

Theorem 1. *Backlog Bound. Let a flow $R(t)$, constrained by an arrival curve α, traverse a system S that offers a service curve β. The backlog $x(t)$ for all t satisfies*

$$x(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\} = v(\alpha, \beta) \tag{10}$$

$v(\alpha, \beta)$ is also often called the vertical deviation between α and β.

Theorem 2. *Delay Bound. Assume a flow $R(t)$, constrained by arrival curve α, traverses a system S that offers a service curve β. At any time t, the virtual delay $d(t)$ satisfies*

$$d(t) \leq \sup_{s \geq 0}\{\inf\{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}\} = h(\alpha, \beta) \tag{11}$$

$v(\alpha, \beta)$ is also often called the vertical deviation between α and β.

As a system theory network calculus offers further results on the concatenation of network nodes as well as the output when traversing a single node. Especially the latter for which now the min-plus deconvolution is used will be of high importance in the sensor network setting as it potentially involves a so-called *burstiness increase* when a node is traversed by a data flow.

Theorem 3. *Output Bound. Assume a flow $R(t)$ constrained by arrival curve α traverses a system S that offers a service curve β. Then the output function is constrained by the following arrival curve*

$$\alpha^* = \alpha \oslash \beta \geq \alpha \tag{12}$$

Theorem 4. *Concatenation of Nodes. Assume a flow $R(t)$ traverses systems S_1 and S_2 in sequence where S_1 offers service curve β_1 and S_2 offers β_2. Then the resulting system S, defined by the concatenation of the two systems offers the following service curve to the flow:*

$$\beta = \beta_1 \otimes \beta_2 \tag{13}$$

Theorem 5. *Blind Multiplexing Nodal Service Curves. Consider a node blindly multiplexing two flows 1 and 2. Assume that the node guarantees a* strict *minimum service curve β and a maximum service $\bar{\beta}$ to the aggregate of the two flows. Assume that flow 2 has α_2 as an arrival curve. Then*

$$\beta_1 = [\beta - \alpha_2]^+$$

is a service curve for flow 1 if $\beta_1 \in \mathcal{F}$. $\bar{\beta}$ remains the maximum service curve also for flow 1 alone. Here, the $[.]^+$ operator is defined as $[x]^+ = \max\{x, 0\}$.