Dimensioning of Time-Critical WSNs – Theory, Implementation and Evaluation

Petcharat Suriyachai[†], Utz Roedig[+], Andrew Scott[+], Nicos Gollan[*], Jens Schmitt[*]
petcharat@coe.psu.ac.th, {u.roedig, a.scott}@lancaster.ac.uk, {gollan, jschmitt}@informatik.uni-kl.de
[†]Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand
[+]InfoLab21, Lancaster University, UK
[*]Distributed Computer Systems Lab, University of Kaiserslautern, Germany

***Abstract*—It is difficult to design and operate a Wireless Sensor Network (WSN) that ensures timely data delivery. Hence, few time-critical WSNs are nowadays in operation despite numerous application scenarios that would benefit from such a system. This paper presents a novel dimensioning framework that can be used to efficiently construct WSNs for time-critical applications. The framework employs the Sensor Network Calculus (SNC) as an analytical method for network dimensioning. A medium access control (MAC) protocol called SNC-MAC is used to ensure that this dimensioning process is reflected accurately in a subsequent deployment. The proposed framework is evaluated using both simulations and measurements in an experimental WSN deployment. The evaluation shows that observed message transfer delays in the deployment never exceed delay bounds determined during network dimensioning. Therefore, the framework can be used to implement WSNs requiring transmission delay guarantees.**

I. Introduction

Future application areas of wireless sensor networks may include industrial process automation, aircraft control systems or traffic management systems. In such systems, the WSN is part of a control loop, and predictable network performance in terms of message transfer delay is required. To construct a functioning control loop, it is often necessary to ensure that sensor data is transported to the decision point within a given time D'. Data arriving late cannot be used in the decision process and has to be considered lost. Early data delivery is acceptable but undesirable as it indicates that the network is overprovisioned to some extent. Overprovisioning should be minimized in a WSN as surplus network capacity consumes scarce energy resources. Thus, an effective tool for precise network dimensioning and deployment is needed.

The Sensor Network Calculus [17] can be used to *dimension* a WSN such that timely data delivery is ensured. The dimensioning process can be decomposed as follows: (a) to choose the number of sensor nodes and the network topology, (b) to set energy-efficiency goals, which are usually expressed as a maximum network lifetime, and (c) to select the sensing and reporting frequency and thus the information resolution.

The SNC is a promising candidate as a dimensioning tool for time-critical WSNs because it allows us to interrelate traffic patterns, network topology, node forwarding capabilities, energy consumption *and* a resulting message transfer delay bound D (with $D \leq D'$). A preliminary study of the SNC dimensioning accuracy [14] shows that SNC predicted worst-case message transfer delay bounds are close to observed delays in *simulations* of WSN scenarios.

However, the SNC has not yet demonstrated its usefulness for WSN dimensioning in *practice*. To apply the SNC in a real deployment, two important questions must be addressed:

1) *How can an SNC dimensioned network be implemented?*
2) *How precise is the SNC dimensioning process?*

The first question must be addressed to ensure that a network described by the SNC can actually be deployed. All assumptions made in the SNC dimensioning process must be represented accurately in a real implementation. For example, the SNC assumes an upper bound for the *node-to-node forwarding latency*, and thus this assumed bound must not be violated in the implementation. Furthermore, the network topology and the routing paths must be kept within the topology envelope assumed in the SNC dimensioning process.

The second question must be addressed to ensure that the dimensioning process does not result in a largely overprovisioned network. If, for example, message transfer delays in real networks are much lower than the worst-case upper bound suggested in the SNC dimensioning process, a waste of resources would occur.

The contribution of of this paper is to provide answers to both questions. To that end, the framework uses the existing SNC as an analytical network dimensioning method. Moreover, the framework employs a novel medium access control protocol called SNC-MAC to ensure that the dimensioning process is reflected accurately in a subsequent deployment.

II. Related Work

The WSN research community has to date produced some partial solutions addressing timely data delivery in sensor networks [22]. Some of these solutions describe the network dimensioning process [1], [3], [8], [17], while others target the construction of real-time capable network components such as MAC protocols [7], [11]. However, these MAC protocols are rather unsuitable as building blocks for time-critical WSN applications because they are not connected with suitable dimensioning tools. In addition, there is currently little work describing the complete process of how to dimension,

Manuscript received February 15, 2011; revised May 15, 2011; accepted June 15, 2011.

doi:10.4304/jcm.6.5.360-369

construct and operate a WSN for time-critical applications. Notable exceptions are discussed next.

Prabh [12] specifies a TDMA-based MAC protocol for constructing a network that is dimensioned using the methods in [1], which are based on scheduling theory. The protocol assumes that a network layout is in a hexagonal shape and only neighboring nodes in the topology interfere. Based on these assumptions, a carefully designed schedule is devised to achieve the minimum possible bound on message transfer delay. Although our framework might result in more pessimistic delay bounds under the assumptions of this work, it exhibits more flexibility with respect to network topology, supported traffic and server models. Moreover, Prabh's methods are evaluated using only simulations.

ANDES [13] is a design tool for wireless sensor networks. A variety of methods are integrated and can be used to analytically analyze network properties before deployment. For example, the aforementioned method in [1] is used in ANDES. However, this tool does not specify how the dimensioned networks can be implemented in practice. ANDES is also evaluated using simulated case studies rather than real deployments. We believe that ANDES is complementary to our work in that the SNC dimensioning method [17] used in this paper could be integrated into ANDES.

PEDAMACS [6] is a TDMA scheme including topology control and routing mechanisms. The sink centrally calculates a transmission schedule for each node, taking interference patterns into account. As a result, an upper bound for the message transfer delay can be analytically determined. Our proposed framework is again more flexible than PEDAMACS in terms of traffic and server models. Furthermore, PEDAMACS is restricted by the requirement of a high-power sink to reach all nodes in the field in a single hop. This protocol is also analyzed using simulations, but a real-world implementation and corresponding measurements are not reported.

RT-Link [15] is a TDMA-based link protocol that creates a connectivity graph and a collision-free slot schedule. The schedule ensures a node using scheduled slots achieves a delay guarantee across multiple hops. RT-Link also relies on special hardware for achieving out-of-band and network wide time synchronization. In contrast, SNC-MAC uses off-the-shelf hardware to avoid additional hardware cost for time synchronization. The protocol exploits existing message transmissions to achieve time synchronization.

The Fieldbus research community has recently investigated the feasibility of using wireless links to replace the traditional wired links in their systems [10]. Although this wireless Fieldbus has properties similar to a WSN with real-time capabilities (see [26] for a review), existing Fieldbus solutions address only one-hop wireless networks or represent mixed wired/wireless solutions [2], [9].

Previous work of ours to support time-critical WSNs consisted in the basic proposal of the SNC [17] which was further advanced to handle in-network processing in [21]. The case of multiple sinks was treated in [20], and the handling of uncertain topologies at dimensioning time was described in [16]. As already mentioned, a preliminary simulation study of the SNC accuracy was performed in [14]. The SNC-MAC detailed in this paper is based on our deterministic MAC protocol described in [23]. In this paper, we extend and tailor the functionality of this protocol such that it meets the SNC dimensioning requirements. Details of these interrelated efforts to establish a WSN dimensioning framework are discussed further in the next section.

III. WSN Dimensioning and Deployment Framework

The framework consists of two components: network dimensioning and network deployment. It adopts the existing SNC for the network dimensioning process and relies on a medium access control protocol called SNC-MAC for the subsequent network deployment. This section explains the SNC and SNC-MAC protocol in detail.

A. *WSN Dimensioning: Sensor Network Calculus*

Any dimensioning process requires some assumptions regarding network structure and operation. The presented framework requires worst-case specifications for network traffic and network topology. Exact information about these parameters may be unavailable in the planning phase of a sensor network, especially in the case of large and random deployments in unknown environments. However, our framework rather targets planned deployments in environments with lower uncertainties. In particular, the number of network nodes is relatively small, and their location can be reasonably determined. Moreover, traffic generation patterns are predictable. An example application scenario which our framework suits well is process monitoring and control in a production plant. In such a setting, the expected traffic can be described, and the network topology can be planned.

The dimensioning process allows us to balance network traffic, topology construction, node forwarding capabilities and energy-efficiency goals such that the required application delay D' can be met. However, an outcome of the dimensioning process might be the conclusion that it is impossible to support an intended application scenario. For instance, the worst-case delay D determined by the SNC may be higher than the required application delay ($D > D'$). In that case, assumptions regarding topology, traffic and node forwarding capabilities must be relaxed to enable application support. In the next section, details of the network calculus are presented as background information before a brief description of the SNC is subsequently provided.

1) Network Calculus Background: Network calculus is a min-plus system theory for deterministic queuing systems that builds on the calculus for network delay in [4], [5]. Network calculus is built around the notion of cumulative functions for input and output flows of data. We denote the set $\mathscr{F}$ of real-valued, non-negative and wide-sense increasing functions passing through the origin as:

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

In particular, the input function $R(t)$ and the output function $R^*(t)$ of a system $\mathscr{S}$ cumulatively count the number of bits that are input to and output from the system.

Definition 1: (Min-plus Convolution and Deconvolution) The min-plus convolution $\otimes$ and deconvolution $\oslash$ of two functions $f,g \in \mathscr{F}$ are defined as

$$(f \otimes g)(t) = \begin{cases} \inf_{0\leq s\leq t}\{f(t-s)+g(s)\} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

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

The min-plus convolution and deconvolution are important operators in the calculus. Let us turn now to the performance characteristics of flows that can be bounded by network calculus means.

Definition 2: (Backlog and Delay) Assume a flow with input function R that traverses a system $\mathscr{S}$, resulting in the output function R^*. The backlog of the flow at time t is defined as

$$B(t) = R(t) - R^*(t).$$

Assuming first-in-first-out delivery, the delay for an input at time t is defined as

$$D(t) = \inf\{\tau \geq 0 : R(t) \leq R^*(t+\tau)\}.$$

Now the arrival and server processes specified by input and output functions are bounded based on the central network calculus concepts of arrival and service curves that are defined next.

Definition 3: (Arrival Curve) Given a flow with input function R, a function $\alpha \in \mathscr{F}$ is an arrival curve for R if

$$\forall t,s \geq 0, s \leq t : R(t) - R(t-s) \leq \alpha(s)$$

$$\Leftrightarrow R \leq R \otimes \alpha \Leftrightarrow \alpha \geq R \oslash R.$$

Definition 4: (Service Curve) Given a flow with input function R that traverses a system $\mathscr{S}$, a function $\beta \in \mathscr{F}$ is a service curve if

$$R^* \geq R \otimes \beta.$$

Theorem 1: (Performance Bounds) Consider a system $\mathscr{S}$ that offers a service curve β and stores input data in a FIFO-ordered queue. Assume a flow R traversing the system that has an arrival curve α. Then we obtain the following performance bounds for the backlog B from the maximum vertical distance $v(\alpha,\beta)$ and for the delay D from the horizontal distance $h(\alpha,\beta)$:

$$\begin{aligned} B(t) &\leq (\alpha \oslash \beta)(0) = v(\alpha,\beta), \\ D(t) &\leq \inf\{t \geq 0 : (\alpha \oslash \beta)(-t) \leq 0\} = h(\alpha,\beta), \\ \alpha^* &\leq \alpha \oslash \beta. \end{aligned}$$

One of the strongest results of network calculus (albeit being a simple consequence of the associativity of $\otimes$) is the concatenation theorem that enables us to investigate tandems of systems as a single system.

Theorem 2: (Concatenation Theorem for Tandem Systems) Consider a flow that traverses a tandem of systems $\mathscr{S}_i$, $i = 1,...,n$. Assume that $\mathscr{S}_i$ offers a service curve β_i, $i = 1,...,n$ to the flow. Then the concatenation of the systems offers a service curve $\bigotimes_{i=1}^{n}\beta_i$ to the flow.

So far we have only covered the tandem network case. The next result factors in the existence of other interfering flows.

Theorem 3: (Left-Over Service Curve) Consider a system that offers a service curve β and that serves two flows R_1,R_2. R_2 is constrained by α_2. Then the left-over service available to R_1 is $\forall t \geq 0$

$$\beta^{l.o.}(t) = \sup_{0\leq s\leq t}\{\beta(s) - \alpha_2(s)\}.$$

This result needs to be generalized to a sink-tree network case in Section III-A2 such that we can apply it in the typical WSN case of sensors reporting towards a single sink.

2) Sensor Network Calculus: The sensor network calculus framework, first described in [17] and further developed in [21], allows a concise worst-case analysis of WSNs. It provides a way to derive bounds on performance measures as, for example, the maximum message transfer delay D experienced by any data flow in a WSN.

a) Sensor Network System Model: We assume the common class of single base station oriented operation models. Although it is possible to accommodate sensor networks with *multiple* sinks as shown in [18], for ease of presentation we restrict on single sink WSNs here. 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 will also receive 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.

b) Incorporation of Network Calculus Components: Now the basic network calculus components, arrival and service curve have to be incorporated. Firstly, 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:

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

As an example, we could use simple token-bucket functions $\gamma_{r,b}(t) = b + rt$ to model inputs.

Secondly, the service curve has to be specified. The service curve depends on the way packets are scheduled in a sensor node, which from the communication perspective, 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. Again as an example, assume a service curve modeling the periodic availability of the full medium capacity C after an initial delay T. This closely captures the characteristics of a TDMA scheme. The structure is similar to the one proposed in [8] for 802.15.4 networks. This curve can be approximated by a so-called rate-latency curve $\beta_{R,T}(t) = \max(R(t-T),0)$.

c) Calculation of Network-Internal Traffic Flow: The output of sensor node *i*, i.e., the traffic that it forwards to its parent in the tree, is constrained by the following arrival curve:

$$\alpha_i^* = \bar{\alpha}_i \oslash \beta_i. \tag{3}$$

Before performance characteristics like the maximum delay from a given node to the sink or local buffer requirements, especially at the most challenged sensor nodes just below the sink (which are called 1-hop nodes in the following), can be calculated, an iterative procedure to calculate the network-internal flows is required as follows:

1) 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 Theorem 1. Each leaf node is now marked as "calculated."
3) For all nodes only having children that are marked "calculated," the output bound α_i^* can be calculated according to Equation (3), and they can again be marked "calculated."
4) If all 1-hop nodes are marked "calculated" the algorithm terminates, otherwise go to step 3.

d) Calculation of Message Transfer Delay Bounds: After the network-internal flows have been computed according to the previous 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:

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

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

One simple way to calculate the message transfer delay bound *D* for a given node is to aggregate the per-node maximum delays on its path to the sink. This method is known as total flow analysis (TFA) because flows are added up on their way to the sink. The TFA method was presented in in [17]. We can improve the TFA by making more careful use of the concatenation theorem, effectively *paying the costs of multiplexing only once*, which results in a method with the name: pay multiplexing only once analysis (PMOOA). The idea of PMOOA is to apply the concatenation as much as possible *before* the application of Theorem 3. The details of this PMOOA method and a discussion of its superiority over the other two methods can be found in [19] and [21].

e) Worst-Case Topologies: The most direct usage of the sensor network calculus assumes a known topology. However, previous work of ours established the concept of worst-case topologies for a specific but typical class of WSN scenarios [16]. The idea in that work is to prove a topology to be the worst with respect to the maximum delay. Under a complete lack of knowledge about the topology, this approach may yield very conservative bounds. Therefore, that work also reasoned about the worst-case topology when at least some restrictions on the topology could be made, in particular due to topology control measures. Here we provide the basic notions and the central result as follows:

Definition 5: (((O,H)-Constrained Tree) A tree is (O,H)-constrained if all of its nodes have an out-degree of less than *O* and none is more than *H* edges away from the root.

Definition 6: (Maximally Deep (O,H)-Constrained Tree) A tree with *n* nodes is a maximally deep (O,H)-constrained tree if it is (O,H)-constrained and the sum of distances from each node to the sink $\sum_{i=1}^{N} e_i$ is maximal. (e_i denotes the number of edges from node n_i to the root), i.e. there is no other (O,H)-constrained tree with a larger sum of distances.

Theorem 4: A sensor network topology that consists of as many nodes as possible below a 1-hop node and has a maximally deep (O,H)-constrained tree below this 1-hop node constitutes a worst case topology with respect to the maximum delay. In addition, this network must consist of *n* nodes in which each node has an arrival curve $\gamma_{r,b}$, and a rate-latency service curve $\beta_{R,T}$ and the network topology is (O,H)-constraint.

B. WSN Deployment: SNC-MAC Protocol

The SNC can only be used in a practical deployment when the network is implemented according to the assumptions made in the SNC dimensioning process. In particular, the assumed worst-case bounds must be obeyed within the implementation. To ensure these bounds, a deterministic node forwarding characteristic (specified by β) and a topology control mechanism that keeps the network structure within the bounds of the assumed maximally deep (O, H)-Constrained Tree are required. These two implementation goals are achieved by defining a specific TDMA-based medium access control protocol called SNC-MAC that provides a deterministic behavior in terms of forwarding delay and includes basic topology control and routing mechanisms.

1) Basic SNC-MAC Operation: All nodes in the network are assumed to be time synchronized. The time axis is divided into fixed-length base units called epochs. Each epoch *E* is subdivided into $m = k \cdot n$ time slots for a network of at most *n* sensor nodes. The time slot has a size of T_s which is large enough to transmit a packet of maximum payload and to receive an acknowledgment from the receiver. Each node is assigned *k* exclusive slots to transmit one message per epoch *E*. Such exclusive slot allocation renders the protocol collision-free, and setting the forwarding rate per epoch results in a deterministic node forwarding characteristic. A node has *k* attempts per epoch to deliver a message to its neighboring receiver, and this message is discarded if all *k* attempts fail. Hence, the *node-to-node latency* is influenced by the value *k* and the epoch duration $E = m \cdot T_s$. Assuming there is no knowledge of slot distribution, in the worst-case scenario the packet is generated after the first transmission slot just passed and successfully sent at the slot that is the last one in the subsequent epoch. Therefore, the node-to-node latency T_n in this worst case is calculated as:

$$T_n \leq E \ \forall k = 1 \text{ and } T_n \leq (2 \cdot E - T_s) \ \forall k > 1$$

2) Routing: The network is formed into a tree structure which is rooted at the sink, and routing is performed by the SNC-MAC. Sensor readings are routed as *sensor data messages* up-tree towards the sink. If necessary, these sensor data messages are queued in the *sensor data FIFO buffer*. In contrast to the sensor data messages, *sink data messages* used to set up and control the sensing tasks can be broadcast from the sink to all nodes in the network. The sink data messages are queued in the *sink data FIFO buffer* if needed, and they are transported when no *sensor data messages* need forwarding. Each node is aware of its position in the tree and knows the slot numbers assigned to its child nodes (to handle sensor data messages) and the parent node (to handle sink data messages). The radio must be in the active state for at least the first slots assigned to the node, its child nodes and its parent node. The node may also be active in additional slots if there is transmission failure in the previous transmission slot. Thus, the topology, the number of transmissions k and the overall number of slots per epoch define the energy consumption pattern of a node.

3) Slot Assignment: The slot assignment of each node can be pre-configured as a planned deployment is assumed. Each node n_i is identified by a unique identification number i with $0 \leq i < n$. Moreover, the node n_i exclusively owns k transmission time slots $s_{i,j}$ with $0 \leq j < k$ in each epoch E. The transmission slots are assigned uniformly across the epoch: $s_{i,j} = i + n \cdot j$. Therefore, retransmission slots obtain the maximum temporal distance which helps to counter burst errors in the channel. However, other types of distribution are acceptable, provided that the slot assignment remains fixed. In addition, regardless of any distribution applied, the epoch length E is constant throughput operation, and consequently the worst-case node-to-node latency T_n is not affected.

4) Error Control: Wireless links have a fluctuating link quality and might even become temporarily unavailable. To deal with this inevitable problem, the SNC-MAC employs two mechanisms: retransmission and skipping. These mechanisms do not violate the SNC assumptions made in the dimensioning process but ensure that message losses can be controlled to stay within acceptable performance bounds under normal operation conditions.

Each node must transmit a packet within its first transmission slot in the epoch. If no sink or sensor data is queued for transmission, a simple *control message* is sent to the parent node for the purpose of link connectivity testing. A packet with a non-broadcast destination address requires an acknowledgment. As mentioned previously, slots are large enough to contain a packet and its subsequent acknowledgment. If a transmission is not acknowledged, a sender will retransmit the message within the next slot of the k transmission slots assigned to it. If a receiver does not receive a message in the expected slot, it will start listening on the next transmission slot assigned to this sender within the epoch. Figure 1 illustrates slot usage of node n_1 in the topology shown in Figure 2 a) when $k = 2$, $n = 20$ and $m = 40$. Slots in which the radio of node n_1 needs to be active are displayed as solid. Slots that might become active in case of packet losses are shown as shaded. Slots in which the radio is always in sleep mode are depicted as empty.

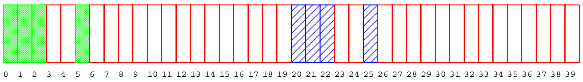


Figure 1. Slot usage of n_1 in Topology A, which is shown in Figure 2 a)

When the link quality becomes poor for an extended period, a node may not be able to successfully transmit a message within the available k transmission slots. The network in this case is temporarily unable to fulfill the application requirements. In addition to signaling to the application that its service is currently disturbed, the MAC protocol will try to correct the problem by adding a transmission path in the topology such that the network dimensioning results remain valid. In particular, if a node does not receive any message during an epoch from a child node, it listens to the transmission slots of grandchildren whose parent is this child node. If transmission range permits, the node can overhear and forward data from these grandchildren. As a result, there is an additional forwarding path that skips a node level and reduces the hop distance. Such topology skipping is acceptable as the resulting tree structure is still within, and even better than, the topology envelope assumed in the SNC dimensioning process. Obviously, the node listening for packets from grandchildren will have to invest additional energy. Moreover, as soon as this node receives a packet from the child node again, the SNC-MAC terminates such topology skipping.

It is important to note again that both error-control mechanisms do not violate the SNC assumptions made in the dimensioning process. They improve data delivery in the reliability domain and do not influence the outcomes of the SNC dimensioning.

5) Energy Consumption: The energy consumption of a node operating SNC-MAC can be calculated before network deployment. This calculation is useful in, for example, industrial deployments where schedules of node maintenance must be predictable.

In general, a duty cycle P (percentage of radio-on-time) of node n_i depends on the number of child nodes c_i attached to it and the epoch length E. Additional energy will be consumed if the aforementioned error-control mechanisms are necessary. A lower bound for the duty cycle of the presented scheme can be given for optimal conditions where these mechanisms do not occur. Specifically, a node n_i is only active in the first of the slots assigned to itself, its parent node and its children. The lower bound for the energy consumption of this node is thus given as:

$$P_i = \frac{T_s \cdot (2 + c_i)}{E} = \frac{2 + c_i}{k \cdot n}$$

An upper bound for the duty cycle can also be given when all retransmission slots and topology skipping are used. In this case, a node n_i is active in all slots assigned to itself, the parent node, child nodes, and slots assigned to the g_i grandchildren. The upper bound for the energy consumption is calculated as:

$$\overline{P_i} = \frac{T_s \cdot k \cdot (2 + c_i + g_i)}{E} = \frac{(2 + c_i + g_i)}{n}$$

For instance, we assume $k = 2$, $n = 20$ and the topology shown in Figure 2 a). Node n_1 in this example has a lower bound for the duty cycle of $\underline{P}_1 = 4/40 = 10\%$ and an upper bound of $\overline{P}_1 = (2+2+4)/20 = 40\%$.

Constant operation at the upper bound would dramatically increase energy consumption and therefore reduce a network lifetime. Two methods can be employed to reduce the energy consumption pattern of the node. First, the topology can be chosen such that the desired energy usage pattern emerges. Second, the epoch length E can be extended by adding additional (unused) slots. The SNC dimensioning process has to be consulted to verify that the desired delay bound can still be maintained while using such methods.

6) Matching Dimensioning Assumptions: The FIFO handling of data traffic combined with the fixed epoch length E allows us to implement the service curve β assumed in the SNC dimensioning. The size E, which defines the upper bound of node-to-node latency T_n, is one of the inputs of the service curve β specification [17]. The network forms an acyclic graph in form of a tree structure that is assumed in the SNC dimensioning process. Moreover, the SNC-MAC protocol always serves data messages from the sensor data FIFO buffer first. Such an operation ensures that data traffic flowing from sensors toward the sink is not disturbed by traffic flowing from the sink to the sensor nodes. As a result, the unidirectional traffic flow modeled in the SNC dimensioning process is correctly reflected in the implementation.

7) Implementation: The SNC-MAC protocol was implemented on TinyOS 2.0.2 for the TelosB [25] platform, which incorporates a CC2420 [24] radio transceiver. Some low-level components had to be modified to obtain the required absolute deterministic behavior. For example, the usage of TinyOS tasks within the MAC component must be avoided as TinyOS cannot guarantee when tasks are executed. Additional details of TinyOS implementation of the protocol are available in [23].

As time synchronization among nodes is necessary, this TDMA-based protocol could be used together with an existing time synchronization method as long as this method would not obstruct data transmission when required. However, to obtain a lightweight implementation, we decided to exploit the already available constant exchange of data and control messages for time synchronization.

Each node in the network synchronizes its clock with the parent node in the tree topology. When a node receives a message (or overhears the message if it is traveling up-tree) generated by its parent, it can calculate the current time in the epoch as it knows which slot in the epoch is used by the parent node. All nodes can use one message in each epoch for time synchronization as each node has to transmit a message in every epoch. Thus, time synchronization is carried out frequently, ensuring an accurate time base in the network. We are aware that our time synchronization method will be problematical for topologies with a long hop distance as synchronization errors will propagate. However, the SNC-MAC protocol is intended to be used in relatively small and controlled deployments. As shown in the evaluation, the described method achieves time synchronization that is sufficiently accurate for these scenarios.

IV. Evaluation Setup

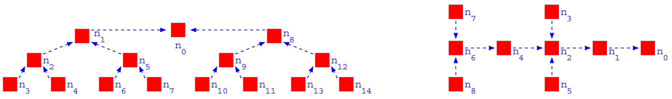
The overall goal of the experimental evaluation was to validate the WSN dimensioning framework against measurements from a real implementation. A testbed using the SNC-MAC implementation for the TelosB platform was used for this purpose. A simulation environment (employing our own event-based simulator) was used to complement the experimental study in order to investigate a larger variety of traffic conditions. The simulator was validated by reproducing the measurement results obtained in the testbed.

The evaluation is carried out using perfect links; in simulation no packets are lost and in the lab deployment packet losses are negligible. Thus, the error-control mechanisms presented in Section III-B are not used in this setup. However, as the error control mechanisms have no impact on delays (see Section III-B), a comparison of SNC calculated delays with the measured delays is feasible in the described evaluation setup.

The application scenarios selected for evaluation represent simple but realistic sensor network settings in which performance control is important. A few sensor nodes ($n \leq 20$) are organized in a tree structure, and messages from all nodes are periodically reported to a single sink. These settings are common for process automation to form a control loop, and the worst-case message transfer delay must not be violated. For example, nodes monitor pressure in pipes of a chemical production plant and forward the measured data to the sink. The sink can decide if an over-pressure situation has occurred and can close a valve to avoid system damage or emergency venting of chemicals. Such a decision has to be made in a timely fashion.

A. Network Topology

Two different network structures: Topology A and Topology B, were used for evaluation and are shown in Figure 2. The first was a simple binary tree with $n = 15$ nodes, while the second was more unbalanced with $n = 8$ nodes. These two structures were chosen to produce very different message transfer delay patterns. We expected that the SNC calculated delay in Topology B should be observed infrequently in a testbed because a long line of nodes with only a few nodes generating cross-traffic would rarely produce the worst possible scheduling conditions. In contrast, we expected message transfer delays close to the SNC calculated worst-case bound would potentially be observed more frequently in Topology A.



a) Topology A b) Topology B

Figure 2. Network Structure

B. Message Forwarding

All nodes implemented the SNC-MAC protocol and forwarded packets according to their specification. The following parameters of the MAC protocol were used for both topologies: $n = 20$, $k = 1$, $m = 20$, $T_s = 9.765ms$ and $E = 20 \cdot T_s = 195.3ms$. It should be noted that TinyOS clocks operate in binary units; $1s$ equals $1024ms$. Therefore, TinyOS time values are not always round numbers. For example, a slot size of $T_s = 9.765ms$ had to be used instead of a $10ms$ value. A packet of size $41bytes$ ($23byte$ payload) was transmitted in a slot, carrying measurement values of interest to the sink. Retransmission was not used ($k = 1$) as the lab based deployment did not experience significant packet losses. Furthermore, the message forwarding in the experimental setup could be described and bounded for the SNC calculations by a rate-latency service curve (see Section III-A2).

C. Network Traffic

The traffic flowed from all sensor nodes toward the sink. Moreover, every node within the network created periodic sensor reports with a frequency of $p\left[\frac{1}{s}\right]$. In a realistic network deployment, these reports may not be generated in each node at exactly the same time. Thus, the experiment time was divided into slots of size $T = 1/p$ duration, and in each slot each node created one message at a randomly selected time between 0 and δ (with $0 \le \delta < T$). In the remainder of the paper δ is called *jitter*.

The jitter parameter δ in the traffic description was used as a variable parameter in the experiments. Changing δ should make it possible to "tune" the experimental setup toward a worst-case behavior. One may expect that more packets close to the SNC bound will be observed for small δ as all nodes become closely synchronized in their transmissions. In contrast, if δ is large, one is tempted to assume that few packets will experience the worst-case delay as the sending activity of the nodes is spread over time and statistical multiplexing effects may be observed.

For the evaluation, time values of $2s \le T \le 3s$ were used. Thus, T allowed us to control the traffic load, while δ allowed us to control how "synchronized" the sensor network is. For small δ, all nodes in the field reported their measurements at roughly the same time toward the sink. This could be the case if, for example, all sensors are set up to report findings periodically. In contrast, with a large δ sensors reported in a more random pattern, which would be the case if, for example, data generation is triggered by locally observed phenomena. The network traffic used in the experimental setup could be described and bounded for the SNC calculations by a maximum sensing rate arrival curve (see Section III-A2).

Table I presents parameters that defined the network traffic in the evaluation setup. The system configuration for both the testbed and simulator were identical. The TelosB nodes were placed closely on a table, and thus a dense network was created. The transmission power was the default TinyOS value of $0dBm$. Each experiment duration was approximately $15min$.

D. SNC Calculated Message Transfer Delay

A tool called *DISCO Network Calculator*[1] [18] was used to perform the SNC calculations necessary to determine the delay bound D. The results of these calculations are shown in Table II and are used for comparison with the measurements taken in the simulation and testbed. We remark the piecewise constant behavior of the calculated delay bounds in the region of interest, which is due to the staircase type of functions used for the modeling of arrival and service curves (see Section III-A).

Parameters	Simulation	Deployment
Traffic Load T increment	$2s \le T \le 3s$ $100ms$	$T = 2s,\ T = 3s$
Jitter δ approximate increment	$0ms \le \delta \le 500ms$ $10ms$	$0ms \le \delta \le 500ms$ $50ms$

Table I
EVALUATION SETUP

Topology	Traffic Load	Worst-Case Delay Bound D
A	$2s \le T \le 3s$	$1.78s$
B	$2s \le T \le 2.5s$ $2.5s < T \le 3s$	$3.76s$ $2.36s$

Table II
SNC WORST-CASE DELAY BOUNDS IN TOPOLOGIES A AND B

The SNC calculation was set up using maximum sensing rate arrivals of one packet every T time units with a one-packet burst and service latency of $185.54ms$, $9.765ms$ slot duration, and a transmission capacity of one packet per slot. This setup resulted in one packet being transmitted every $E = 195.3ms$. The SNC analysis was performed for both topologies for $2s \le T \le 3s$ to evaluate SNC worst-case prediction accuracy for a variety of traffic loads.

It has to be noted that the traffic with inter-arrival times below a threshold T_{min} cannot be supported as the network would be overloaded, resulting in infinite delay bounds. In Topology A this threshold was $T_{min} = 1.367s$, while in Topology B the threshold was $T_{min} = 1.661s$. The SNC analysis can hence be used before a network deployment to determine the maximum network capacity defined by T_{min}. Note that the jitter parameter δ is not an input in this analysis as the SNC arrival curve already represented the maximum traffic for all possible values of δ. The jitter parameter serves a purpose in the evaluation as it produces different traffic scenarios, which consequently enable us to test the effectiveness of the SNC dimensioning framework.

If the network is dimensioned according to the setup described above, all messages can be delivered within the SNC delay bounds ($d_{max} \le D$). If the application can deal with the delay bound D, the sensor network is dimensioned correctly ($D < D'$). If the SNC delay bounds are significantly lower than the required application delay ($D \ll D'$), the forwarding capabilities of the nodes can be decreased, which would result in a lower duty cycle and thus energy savings. In contrast, if the calculated delay is higher than the required delay ($D > D'$), a different topology can be selected or the forwarding rate can be increased if possible, for example, by reducing the epoch length E of the MAC layer. This way, a resource dimensioning of the WSN can effectively be performed.

[1] http://disco.informatik.uni-kl.de/content/Network_Calculator

V. Evaluation Results

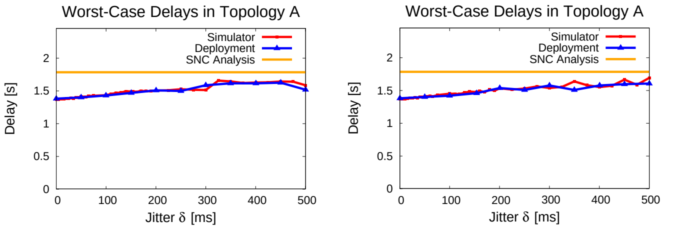
In order to validate and evaluate the accuracy of the SNC dimensioning process, we need to know (1) *whether all measured delays are below the SNC calculated worst-case bound* and (2) *how close the measured delays are to the SNC bound*.

The first question can simply be answered by comparing the SNC calculated delay bound with the delays observed in the experiment. Moreover, a *dimensioning accuracy* P is defined and used to identify the percentage of how much the measured worst-case delays are below the SNC bound. Formally, $P = (D - d_{max})/D$ with D as the SNC calculated delay bound and d_{max} as the worst-case delay observed in a particular measurement or simulation trace.

In contrast, to answer the second question, a more complex analysis is required. Besides determining the absolute difference between a measured worst-case delay and a calculated SNC delay bound, it is necessary to analyze the delay distribution. To explain this distribution, a measure is needed to quantify how many messages actually experience a delay that can be considered close to the SNC bound. We decided to use a measure called *effective window* W which describes the number of messages that experience a delay of more than half of the SNC bound. Formally, $W = M_{half}/M$ with M being the number of all messages and M_{half} being the number of messages that are above $\frac{1}{2}D$. This section presents the evaluation results using the aforementioned measures.

A. Measured Message Transfer Delay in Topology A

All measured delays observed in this deployment are below the SNC bound of $1.78s$ as expected. This result validates that the framework using the SNC for dimensioning and the SNC-MAC for deployment functions correctly.



a) $T = 2s$ and $0ms \leq \delta \leq 500ms$ b) $T = 3s$ and $0ms \leq \delta \leq 500ms$

Figure 3. Comparison of worst-case delays in Topology A. Delays are obtained from simulator, deployment and SNC analysis.

Figure 3 shows the worst-case delays obtained from the testbed, the simulator and the SNC calculation for two traffic patterns defined by $T = 2s$ with $(0ms \leq \delta \leq 500ms)$ and $T = 3s$ with $(0ms \leq \delta \leq 500ms)$. The x-axis displays the jitter values $(0ms \leq \delta \leq 500ms)$ used in the evaluation, while the y-axis depicts the corresponding worst-case delays. The comparison between the testbed and simulation results suggests that the simulation environment accurately reflects the conditions in the testbed as both curves match closely. Thus, it is justified to use the simulator to evaluate additional traffic conditions. More simulation results are presented in Figure 4. The x-axis shows the simulation runs for different δ $(0ms \leq \delta \leq 500ms)$.

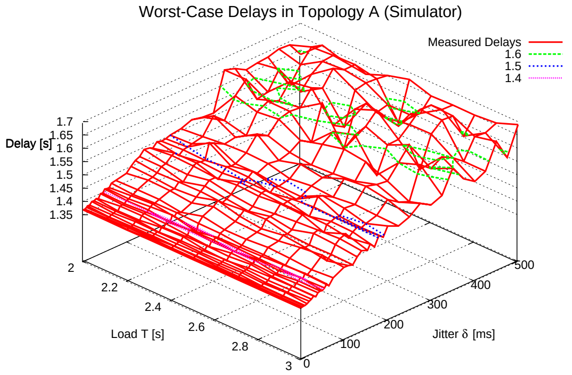


Figure 4. Measured worst-case delays in Topology A. Delays are obtained by simulation for $2s \leq T \leq 3s$ and $0ms \leq \delta \leq 500ms$.

The y-axis depicts the different traffic patterns given by T $(2s \leq T \leq 3s)$. The z-axis illustrates the measured worst-case delay in the simulation environment. This figure emphasizes that the simulation can minimize the risk of missing a delay maximum for a traffic setting not covered by the testbed experiments. Hence, the presented results from both the simulation and testbed are adequate for evaluating the framework.

In addition, the graphs in Figure 3 and Figure 4 illustrate that the worst-case delay increases slightly when the jitter δ is incremented. In the case $\delta = 0$, all nodes in the field periodically generate a message at exactly the same time, and consequently forwarding queues in the network are filled in a deterministic manner. However, in the case $\delta > 0$, all nodes have an offset when generating messages, and the measured worst-case delay is higher than that for $\delta = 0$. Thus, the fully synchronized network with $\delta = 0$ does not represent - as one might think - the worst possible scenario. Moreover, a jitter $(\delta > 0)$ in message generation patterns enables unfavorable scheduling situations as a high worst-case delay can still occur at a large jitter value.

The smallest distance between a measured worst-case delay in the testbed and the SNC predicted worst-case delay can be found for a traffic setting of $T = 2s$ and $\delta = 450.195ms$. At this point, the measured worst-case delay of $d_{max} = 1.621s$ is just 8.93% below the SNC predicted worst-case delay of $D = 1.78s$, i.e., achieving a dimensioning accuracy of $P = 8.93\%$. The simulation, investigating additional traffic settings, reveals a setting with even a better dimensioning accuracy of $P_{max} = 4.69\%$ at $T = 2.3s$ and $\delta = 425ms$. Moreover, the worst prediction accuracy of $P_{min} = 23.31\%$ is observed in the simulation. A summary of the dimensioning accuracy values is shown in Table III.

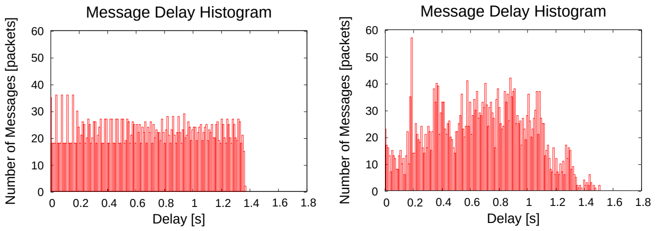
Figure 5 presents detailed delay histograms for the testbed experiments with two different jitter values δ and a traffic load of $T = 2s$. When comparing the number of messages with the delays in the range of $1.3s \leq d_{max} \leq 1.78s$ for $\delta = 0ms$ and $\delta = 500ms$, there are more messages that experience high delays d_{max} close to the SNC bound of $D = 1.78s$ for $\delta = 500ms$. This result thus confirms that the jitter is essential in the evaluation as it can generate worst-case traffic scenarios, which are infrequent but crucial, to test the precision of the

		Simulation	Deployment
Best Prediction	P_{max}	4.69%	8.93%
	d_{max}	$1.696s$	$1.621s$
	(δ, T)	$(425ms, 2.3s)$	$(450.195ms, 2s)$
Worst Prediction	P_{min}	23.31%	22.65%
	d_{max}	$1.365s$	$1.377s$
	(δ, T)	$(2ms, 2.2s)$	$(0ms, 3s)$

		Simulation	Deployment
	W_{max}	35.44%	35.67%
	(δ, T)	$(55ms, 2.3s)$	$(99.609ms, 3s)$
	W_{min}	27.26%	29.81%
	(δ, T)	$(500ms, 2.5s)$	$(500ms, 3s)$

Table III
DIMENSIONING ACCURACY AND EFFECTIVE WINDOW IN TOPOLOGY A.

SNC prediction. In addition, the histograms show that long message delays are not rare events, and the effective window can capture such behavior. In the testbed, the effective window can be as large as $W_{max} = 35.67\%$, whereas the smallest effective window is 29.81%. The largest and smallest effective windows encountered in the evaluation are listed in Table III.



a) $\delta = 0ms$ b) $\delta = 500ms$

Figure 5. Delay histograms for Topology A, $T = 2s$. Measurements are obtained from the deployment.

B. Measured Message Transfer Delay in Topology B

		Simulation	Deployment
Best Prediction	P_{max}	19.85%	20.55%
	d_{max}	$1.891s$	$1.875s$
	(δ, T)	$(400.39ms, 2.8s)$	$(500ms, 3s)$
Worst Prediction	P_{min}	58.51%	58.44%
	d_{max}	$1.560s$	$1.563s$
	(δ, T)	$(2ms, 2.2s)$	$(0ms, 2s)$

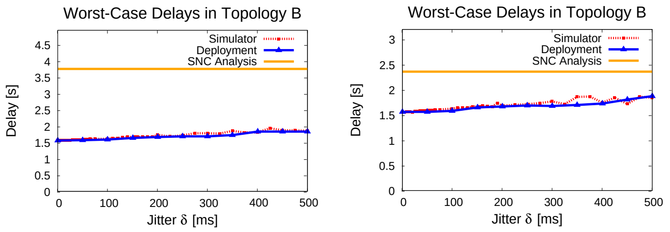
		Simulation	Deployment
	W_{max}	27.5%	26.52%
	(δ, T)	$(299.804ms, 2.9s)$	$(299.804ms, 3s)$
	W_{min}	0%	24.51%
	(δ, T)	$(\delta, T \leq 2.5s)$	$(500ms, 3s)$

Table IV
DIMENSIONING ACCURACY AND EFFECTIVE WINDOW IN TOPOLOGY B.

All measured delays in the experimental runs are below the SNC bounds of $3.76s$ for $(2s \leq T \leq 2.5s)$ and $2.36s$ for $(2.5s < T \leq 3s)$. Table IV shows the range of the observed worst-case delays d, dimensioning accuracy P and effective window W.

For the experimental runs with $2s \leq T \leq 2.5s$, the effective window is empty; no message experiences a delay of at least 50% of the SNC bound. In this particular case, the SNC bound is relatively far away from the experienced delays. Since the duration of each experiment run was only $15min$, this result was expected to be observed rarely in this topology as described in Section IV. However, the result deserved further investigation. We therefore ran one extra simulation experiment with $T = 2s$ and $\delta = 1.0s$ for a period of $11.5days$ in simulation time. In this much longer run, we measured a worst-case delay of $d_{max} = 2.43s$ which is 35.4% below the SNC bound. Hence, delays closer to the SNC bound can occur, but they represent increasingly rare events within this network configuration.

Figure 6 and Figure 7 depict a graphical summary of all experimental runs. Similar to the results for Topology A, the worst-case delay is slightly dependent of the jitter δ, and the simulation results and testbed measurements match closely. The maximum deviations between worst-case delays obtained in the simulation and testbed are 7.51% ($\delta = 349ms$) and 9.48% ($\delta = 349ms$) for $T = 2s$ and $T = 3s$, respectively.



a) $T = 2s$ and $0ms \leq \delta \leq 500ms$ b) $T = 3s$ and $0ms \leq \delta \leq 500ms$

Figure 6. Comparison of worst-case delays in Topology B. Delays are obtained from simulator, deployment and SNC analysis.

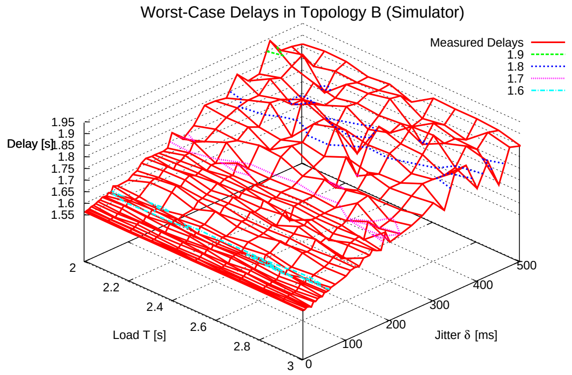


Figure 7. Measured worst-case delays in Topology B. Delays are obtained by simulation for $2s \leq T \leq 3s$ and $0ms \leq \delta \leq 500ms$.

C. Findings

All worst-case delays measured in the simulation and testbed are *below* the SNC bounds in both topologies. Table V presents the SNC prediction accuracy in the most favorable and unfavorable cases. The *best* SNC prediction is as close as 4.69% for Topology A, while the worst prediction occurs in Topology B as expected. Moreover, the *effective window* W emphasizes the importance of the SNC dimensioning. For example, W_{max} of delays observed in the simulation in Topology A is 35.44%. Therefore, about a third of the messages would be discarded if a provisioning error of 50% is made.

	Simulation $2s \le T \le 3s$	Deployment $T = 2s,\ T = 3s$
Best, P $(Topology, \delta, T)$	4.69% $(A, 425ms, 2.3s)$	8.93% $(A, 450.195ms, 2s)$
Worst, P $(Topology, \delta, T)$	58.51% $(B, 2ms, 2.2s)$	58.44% $(B, 0ms, 2s)$

Table V
SUMMARY OF THE SNC PREDICTION ACCURACY

The simulation represents an perfect channel without transmission losses that nevertheless cannot be achieved in practice. Within all testbed experiments, a maximum packet loss rate of 0.29% is observed in Topology B. As the rate is very low, it has no measurable impact on the experienced worst-case delays. Such performance correctly signifies the dimensioning process that does not use redundant slots. Additionally, the lower bound for the energy consumption of a node explained in Section III-B is achieved.

In summary, the evaluation of the simulation and testbed validates that the proposed framework can be used to dimension and construct a WSN effectively. It is also important to note that the presented dimensioning is based on *normal* operating conditions. Unexpected conditions such as signal jamming would clearly invalidate all dimensioning efforts.

VI. CONCLUSION

This paper presents a framework for resource dimensioning of WSNs for time-critical applications. The framework utilizes Sensor Network Calculus (SNC) as an analytical foundation to calculate the worst-case message transfer delay of a WSN using traffic pattern, network topology and energy saving regimes as input parameters. As the SNC makes assumptions on the worst-case forwarding behavior of nodes, the framework employs the SNC-MAC protocol to ensure that these assumptions are reflected accurately in a network deployment.

The proposed framework is evaluated using both simulations and a testbed. The results illustrate that the SNC predictions for message transfer delays are reasonably close to the actually observed values in the simulations and testbed for a balanced tree topology. Furthermore, most importantly for time-critical applications, the SNC bounds are never violated in both deployed topologies. Therefore, the SNC together with suitable building blocks like the SNC-MAC protocol presented in this paper is a promising candidate for safe dimensioning of WSNs for time-critical applications.

Acknowledgement

This work has been partially supported by the European Commission under the FP7 contract FP7-ICT-224282 (GINSENG).

REFERENCES

[1] T. Abdelzaher, S. Prabh, R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *Proc. 25th IEEE Int. Real-Time Systems Symp.*, Lisbon, Portugal, 2004, pp. 359-370.

[2] M. Alves, E. Tovar, F. Vasques, G. Hammer, K. Röther, "Real-time communications over hybrid wired/wireless PROFIBUS-based networks," in *Proc. 14th Euromicro Conf. on Real-Time Systems*, Vienna, Austria, 2002, pp. 142-151.

[3] C.-F. Chiasserini, M. Garetto, "Modeling the performance of wireless sensor networks," in *Proc. 23rd Annu. Joint Conf. IEEE Computer and Communications Societies*, Hong Kong, 2004.

[4] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, pp. 114-131, 1991.

[5] R. L. Cruz, "A calculus for network delay, Part II: Network analysis," *IEEE Trans. Inf. Theory*, vol. 37, pp. 132-141, 1991.

[6] S. Ergen, P. Varaiya, "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks," *IEEE Trans. Mobile Comput.*, vol. 5, pp. 920-930, Jul. 2006.

[7] G. Halkes, K. Langendoen, "Crankshaft: An energy-efficient MAC-protocol for dense wireless sensor networks," in *Proc. 4th European Conf. Wireless Sensor Networks*, The Netherlands, 2007, pp. 228-244.

[8] A. Koubaa, M. Alves, E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proc. 27th IEEE Int. Real-Time Systems Symp.*, Rio de Janeiro, Brazil, 2006, pp. 412-421.

[9] C. Koulamas, S. Koubias, G. Papadopoulos, "Using cut-through forwarding to retain the real-time properties of Profibus over hybrid wired/wireless architectures," *IEEE Trans. Ind. Electron.*, vol. 51, pp. 1208-1217, Dec. 2004.

[10] K. Koumpis, L. Hanna, M. Andersson, M. Johansson, "Wireless Industrial Control and Monitoring beyond Cable Replacement," in *Proc. 2nd PROFIBUS Int. Conf.*, Coombe Abbey, Warwickshire, UK, 2005.

[11] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. 18th Int. Parallel and Distributed Processing Symp.*, Santa Fe, NM, USA, 2004, pp. 224-231.

[12] K. Prabh, "Real-Time Wireless Sensor Networks," Ph.D. dissertation, Dept. Computer Science, Univ. Virginia, VA, USA, 2007.

[13] V. Prasad, et al., "ANDES: An Analysis-based Design Tool for Wireless Sensor Networks," in *Proc. 28th IEEE Int. Real-Time Systems Symp.*, Tucson, AZ, USA, 2007, pp. 203-213.

[14] U. Roedig, N. Gollan, J. Schmitt, "Validating the Sensor Network Calculus by Simulations," in *Proc. 2nd Performance Control in Wireless Sensor Networks Workshop*, Austin, Texas, USA, 2007.

[15] A. Rowe, R. Mangharam, R. Rajkumar, "RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks," in *Proc. 3rd Annu. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, Reston, VA, USA, 2006, vol. 2, pp. 402-411.

[16] J. Schmitt, U. Roedig, "Worst-Case Dimensioning of Wireless Sensor Networks under Uncertain Topologies," in *Proc. 1st Workshop on Resource Allocation in Wireless Networks*, Riva del Garda, Italy, 2005.

[17] J. Schmitt, U. Roedig, "Sensor Network Calculus - A Framework for Worst-Case Analysis," in *Proc. 1st IEEE Int. Conf. Distributed Computing in Sensor Systems*, Marina del Rey, CA, USA, 2005, pp. 141-154.

[18] J. Schmitt, F. Zdarsky, "The DISCO Network Calculator - A Toolbox for Worst-Case Analysis," in *Proc. 1st Int. Conf. on Performance Evaluation Methodologies and Tools*, Pisa, Italy, 2006.

[19] J. Schmitt, F. Zdarsky, M. Fidler, "Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch ...," in *Proc. 27th IEEE Conf. Computer Communications*, Phoenix, AZ, USA, 2008, pp. 1669-1677.

[20] J. Schmitt, F. Zdarsky, U. Roedig, "Sensor Network Calculus with Multiple Sinks," in *Proc. Performance Control in Wireless Sensor Networks Workshop, in conjunction with 2006 IFIP Networking Conf.*, Coimbra, Portugal, 2006, pp. 6-13, ISBN 972-95988-5-1.

[21] J. Schmitt, F. Zdarsky, L. Thiele, "A Comprehensive Worst-Case Calculus for Wireless Sensor Networks with In-Network Processing," in *Proc. 28th IEEE Int. Real-Time Systems Symp.*, Tucson, AZ, USA, 2007, pp. 193-202.

[22] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha, J. Hou, "Real-time communication and coordination in embedded sensor networks," *Proc. IEEE*, vol. 91, pp. 1002-1022, Jul. 2003.

[23] P. Suriyachai, U. Roedig, A. Scott, "Implementation of a MAC Protocol for QoS Support in Wireless Sensor Networks," in *Proc. 1st Int. Workshop Information Quality and Quality of Service for Pervasive Computing, in conjunction with 7th Annu. IEEE Int. Conf. Pervasive Computing and Communications*, Galveston, TX, USA, 2009, pp. 1-6.

[24] TI/Chipcon CC2420 Datasheet, [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc2420.pdf, Feb. 2011.

[25] Tmote Sky Datasheet, [Online]. Available: http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf, Feb. 2011.

[26] A. Willig, K. Matheus, A. Wolisz, "Wireless Technology in Industrial Networks," *Proc. IEEE*, vol. 93, pp. 1130-1151, Jun. 2005.