Towards Predictable Wireless Sensor Networks – The Sensor Network Calculus

Jens B. Schmitt

disco | Distributed Computer Systems Lab, University of Kaiserslautern, Germany

Abstract. At the current state of affairs it is hard to obtain a predictable performance from wireless sensor networks, not to mention performance guarantees. In particular, a widely accepted and established methodology for modeling the performance of wireless sensor networks is missing. In the last two years we have tried to make a step into the direction of an analytical framework for the performance modeling of wireless sensor networks based on the theory of network calculus, which we customized towards a so-called *sensor network calculus* [1]. We believe the sensor network calculus to be especially useful for applications which have certain timing requirements. Examples for this class of applications are factory control, nuclear power plant control, medical applications, and any alerting systems. In general whenever the sensed input may necessitate immediate actions the sensor network calculus may be the way to go. In this paper we summarize these activities and discuss the open issues for such a an analytical framework to be widely accepted.

1 Introduction

Decisions in daily life are based on the accuracy and availability of information. Sensor networks can significantly improve the quality of information as well as the ways of gathering it. For example, sensor networks can help to get higher fidelity information, acquire information in real time, get hard-to-obtain information, and reduce the cost of obtaining information. Application areas for sensor networks might be production surveillance, traffic management, medical care, or military applications. In these areas it is crucial to ensure that the sensor network is functioning even in a worst case scenario. If a sensor network is used for example for production surveillance, it must be ensured that messages indicating a dangerous condition are not dropped. If functionality in worst case scenarios cannot be proven, people might be in danger and the production system might not be certified by authorities.

As it may be difficult or even impossible to produce the worst case in a real world scenario or in a simulation in a controlled fashion, an analytical framework is desirable that allows a worst case analysis in sensor networks. Network calculus [2] is a relatively new tool that allows worst case analysis of packet-switched communication networks. In [1] a framework for worst case analysis of wireless sensor networks based on network calculus is presented and called *sensor network calculus*. This framework has further been extended towards random deployments [3] and the case of multiple sinks in [4]. The goal of this paper is to summarize these activities and show the usefulness of the sensor network calculus as well as opportunities for future work along this avenue.

2 Sensor Network Calculus: A Brief Walk-Through

In this section we use the notation and the basic results provided in [1], furthermore a single sink communication pattern is assumed. 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.

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 childrens' input, is:

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

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

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

2.1 Maximum Sensing Rate Arrival Curve

The simplest option in bounding the sensing input at a given sensor node is based on its maximum sensing rate which is either due to the way the sensing unit is designed or limited to a certain value by the sensor network application's task in observing a certain phenomenon. For example, it might be known that in a temperature surveillance sensor system, the temperature does not have to be reported more than once per second at most. The arrival curve for a sensor node i corresponding to simply putting a bound on the maximum sensing rate is

$$\alpha_i(t) = p_i t = \gamma_{p_i,0}(t) \tag{3}$$

This arrival curve can be used in situations where all sensor nodes are set up to periodically report the condition in a sensor field. The set of sensible arrival curve candidates is certainly larger than the arrival curves described above. The more knowledge on the sensing operation and its characteristics is incorporated into the arrival curve for the sensing input the better the performance bounds become.

2.2 Rate-Latency Service Curve

Next, 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 specific, the service curve depends on how the duty cycle and therefore the energy-efficiency goals are set.

The service curve captures the characteristics with which sensor data is forwarded by the sensor nodes towards the sink. It abstracts from the specifics and idiosyncracies of the link layer and makes a statement on the minimum service that can be assumed even in the worst case. A typical and well-known example of a service curve from traditional traffic control in a packet-switched network is

$$\beta_{R,T}(t) = R\,[t - T]^{+} \tag{4}$$

where the notation $[x]^{+}$ equals x if $x \geq 0$ and 0 otherwise. This is often also called a rate-latency service curve. The latency term nicely captures the characteristics induced by the application of a duty cycle concept. Whenever the duty cycle approach is applied there is the chance that sensed data or data to be forwarded arrives after the last duty cycle (of the next hop!) is just over and thus a fixed latency occurs until the forwarding capacity is available again. In a simple duty cycle scheme this latency would need to be accounted for for all data transfers. For the forwarding capacity it is assumed that it can be lower bounded by a fixed rate which depends on transceiver speed, the chosen link layer protocol and the duty cycle. So, with some new parameters the following service curve at sensor node i is obtained:

$$\beta_i(t) = \beta_{f_i,l_i}(t) = f_i[t - l_i]^{+} \tag{5}$$

Here f_i and l_i denote the forwarding rate and forwarding latency for sensor node i.

2.3 Network Flow Analysis

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:

$$\alpha_i^{*} = \bar{\alpha}_i \oslash \beta_i = \left(\alpha_i + \sum_{j=1}^{n_i} \alpha^{*}_{child(i,j)}\right) \oslash \beta_i \tag{6}$$

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) an iterative procedure to calculate the network internal flows is required:

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 (6). 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 (6) 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 per node delay bounds D_i for each sensor node i can be calculated according to a basic network calculus result [2, chapter 1]:

$$D_i = h(\bar{\alpha}_i, \beta_i) = \sup_{s \geq 0}\{\inf\{\tau \geq 0 : \bar{\alpha}_i(s) \leq \beta_i(s + \tau)\}\} \quad (7)$$

To compute the total information transfer delay $\bar{D}_i$ for a given sensor node i the per node delay bounds on the path $P(i)$ to the sink need to be added:

$$\bar{D}_i = \sum_{i \in P(i)} D_i \quad (8)$$

The maximum information transfer delay in the sensor network can then obviously be calculated as $\bar{D} = \max_{i=1,\ldots,N} \bar{D}_i$. Note that this kind of analysis assumes FIFO scheduling at the sensor nodes which however should be the case in most practical cases.

3 Advanced Sensor Network Calculus

After this brief walk-through the sensor network calculus basics, we will discuss some of the more advanced techniques we have developed to further customize network calculus to the wireless sensor network setting as well as some of the applications of the framework we have proposed.

We have seen in the previous section how the single sink communication pattern typically found in wireless sensor networks was used to iteratively work out the internal traffic flow bounds inside the network and use these to calculate delay bounds in an additive fashion. However, one of the strengths of network calculus is its powerful concatenation result which allows in general to achieve better bounds when a tandem of servers is first min-plus convoluted to a single system compared to an additive analysis of the separate servers. This concatenation result is not directly applicable in a wireless sensor network scenario even when only considering the simple single sink case. Therefore, we have generalized the concatenation result for general feedforward networks in [5], introducing a principle called "pay multiplexing only once" which makes optimal use of subpaths shared between flows and achieves improvements over the additive bounds which may be on the order of magnitudes depending on the scenario. A further extension of the basic sensor network calculus which we also describe in [5] is the integration of maximum service curves into the sensor network calculus which allows to improve the bounds on the network-internal flows and thus in turn lowers the performance bounds, again often very considerably. All these techniques, among other general network calculus techniques, have been implemented in the so-called DISCO Network Calculator. As we believe that tool support is of great importance for a wide acceptance of the sensor network calculus we provide the DISCO Network Calculator in the public domain[1].

Apart from trying to push the sensor network calculus forward methodwise, we have also illustrated how to apply it for several design and control issues in wireless sensor networks. In [1] we have shown how a buffer dimensioning of the sensor nodes may be performed based on the worst case analyses of sensor network calculus such that no information is lost due to buffer overflow inside the network. Furthermore, we also discussed in [1] how different choices of duty cycles affect the information transfer delay. In [3], we considered the case of a randomly deployed sensor network and how this further dimension of uncertainty can be factored into the sensor network calculus. In particular we discussed how constraints from topology control may be used to improve the performance bounds from the sensor network calculus. Thus we proposed to guide topology control decisions based on the sensor network calculus models. In [4] we used the advanced sensor network calculus result discussed in the previous paragraph to investigate scenarios with multiple sinks. In particular we demonstrated how sensor network calculus can be used to dimension the number of sinks as well as their placement in the sensor field.

4 Open Issues and Future Work Items

While we believe the sensor network calculus to have potential, there are still many open issues and correspondingly opportunities for future work. One immediate issue arising from the use of a deterministic analytical framework is the question how to capture the inherently stochastic nature of wireless communications. Here, we plan to integrate lately upcoming stochastic extensions of network calculus [6], which however again need to be customized for the sensor network case. Another issue is how to take in-network processing as is frequently proposed for wireless sensor networks into account. In [7] we have proposed a network calculus that allows for the scaling of data flows. This development should enable modelling of typical in-network processing techniques as for example aggregation of information. Furthermore, it should also be possible to accomodate the mobility of sensor nodes and/or sinks. As in many scenarios this is a kind of controlled mobility there is hope to capture even this difficult characteristic of advanced wireless sensor network scenarios.

Apart from these fundamental issues for the sensor network calculus, it is also important to demonstrate its usefulness in further applications. At the moment

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

we design a task admission control scheme based on sensor network calculus for sensor networks that may have several concurrent tasks. Another work item could be a scheme where sleeping nodes are activated such that certain performance bounds can still be satisfied. Apart from these issues the presented framework should also be validated by packet-level simulations in order to increase the fidelity in the predictive power of our models. Especially this last point deserves our immediate attention and is already currently under investigation.

References

1. Jens Schmitt and Utz Roedig. Sensor Network Calculus - A Framework for Worst Case Analysis. In Proc. of IEEE/ACM Int. Conf. on Distributed Computing in Sensor Systems (DCOSS'05), pages 141-154. Springer, LNCS 3560, June 2005.
2. J.-Y. Le Boudec and P. Thiran. Network Calculus - A Theory of Deterministic Queueing Systems for the Internet. Springer, LNCS 2050, 2001.
3. Jens Schmitt and Utz Roedig. Worst Case Dimensioning of Wireless Sensor Networks under Uncertain Topologies. In Proc. of 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05), Workshop on Resource Allocation in Wireless Networks, Riva del Garda, Italy. IEEE, April 2005.
4. Jens B. Schmitt, Frank A. Zdarsky, and Utz Roedig. Sensor Network Calculus with Multiple Sinks. In Proc. of IFIP Networking 2006, Workshop on Performance Control in Wireless Sensor Networks, Coimbra, Portugal. Springer LNCS, May 2006.
5. Jens B. Schmitt, Frank A. Zdarsky, and Ivan Martinovic. Performance Bounds in Feed-Forward Networks under Blind Multiplexing. Technical Report 349/06. University of Kaiserslautern, Germany, April 2006.
6. Yuming Jiang. A Basic Stochastic Network Calculus. In Proc. ACM SIGCOMM 2006, Pisa, Italy, September 2006.
7. Markus Fidler and Jens B. Schmitt. On the Way to a Distributed Systems Calculus: An End-to-End Network Calculus with Data Scaling. In Proc. ACM SIGMETRICS/Performance 2006 (SIGMETRICS'06), St. Malo, France, June 2006.