Model-Based QoS Evaluation for Embedded Wireless Sensor Networks

Sven Jäger, Tino Jungebloud, Ralph Maschotta, and Armin Zimmermann
System and Software Engineering Group
Faculty of Computer Science and Automation
Ilmenau University of Technology
Ilmenau, Germany
Email: See http://www.tu-ilmenau.de/sse

***Abstract*—Wireless sensor network technologies can be used for industrial embedded system design to save cabling cost and weight. However, their environment, architecture, and design issues are quite different from original sensor network applications. Simulation tools and system design environments have to be adapted to the special requirements, such as operational modes and phases which influence the system. This paper presents a more detailed wireless sensor network to check against the application requirements concerning QoS, reliability, and lifetime. An avionics application example is presented.**

I. Introduction

Modern embedded systems are often a combination of heterogeneous distributed processing units as well as various sensors and actuators. The amount of cabling for their communication is thus increasing together with added functionality, which is problematic for their lightweight and cost-efficient design and leads to costly maintenance and inflexibility against configuration changes.

Wireless sensor network technology (WSN) may overcome this drawback [1] [2]. Unfortunately, applied research is primarily focused on dispersed sensing in unknown environments with subjects like topology estimation or multi-hop packet transmission. Industrial embedded products in aerospace and automobile sectors, the situation is quite different - network topology is pre-planned and well known. However, WSN technologies such as independent sensor nodes as well as energy management and harvesting can be exploited. During such a technology change, it is not obvious how the requirements on reliability, latency, etc. of the applications running on it can be assured wirelessly. On the other hand, these non-functional properties are of much higher importance for real-time embedded systems than for standard WSNs.

WSNs have numerous design parameters which can have a major impact on operation and non-functional properties. Their system design thus benefits greatly from a model-based analysis. Industrial embedded systems such as vehicles or aircraft are characterized by operational phases (modes) with significantly varying system load and environmental factors. Simulation studies of WSNs are often carried out under simplified worst-case assumptions [3] and in unrealistic simple scenarios with homogenous traffic requirements. Those analyses results differ significantly from real scenarios with respect to data traffic, energy consumption, and lifetime. In addition to that, it is not possible to analyze transient behavior at mode changes, which is much harder than a worst-case deterministic evaluation showing that there is no overload in a stationary environment.

This paper presents a more detailed modeling and simulation-based analysis approach for industrial WSN design in an aircraft cabin environment. Time-dependent environmental changes are taken into account, thus leading to a significantly improved prediction of energy consumption, lifetime of sensor nodes, data traffic, and error rate especially under changing environmental and operational aspects.

There are several kinds of QoS parameters in WSNs. An overview of these parameters where given by Dazhi Chen [4]. In [5] they validate QoS parameters for a biomedical sensor network with UPAAL, a tool for modeling and validating of real-time systems which are modeled as networks of timed automata [6].

There are several well developed networks simulators which were extended to simulate WSN. For example the open source simulators NS-2 [7], OMNETT++ [8] and the commercial OPNET [9], which based on discrete event simulation.

The paper is structured as follows. The generic structure of our model is given in Section II. Go ahead with some implementation details and show and analyze a real world example with two applications.

II. A Model of Industrial Embedded Sensor Networks

Our model consists of two components, the environment and the WSN network itself. The environment generates input for the network model. It consists of three parts: the scheduler, which defines a sequence of operation phases with their duration. The second one is the traffic generator, which controls the traffic load generated by sensor nodes. It is possible to assign different traffic patterns to each node depending on the application and the current operation phase. The third part describes energy consumption and generation, which calculates the amount of energy a sensor node can obtain depending on its physical position. This could be influenced by the current phase as well.

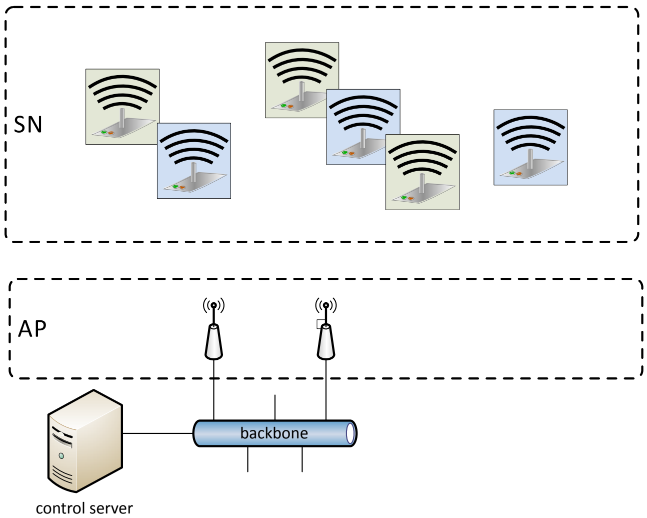

Fig. 1. Top-level view of WSN model

The network model is in our avionic application a simplified wireless sensor network with a centralized control server. Its top level is depicted in Figure 1 and consists of central control server, backbone network, cloud of access points (AP), shared medium and a set of sensor nodes (SN). Event channels are used to transport packages between the components. The components communicate over a shared medium, which may lead to interferences of different kinds. For the communication, three different kinds of protocols have been implemented: a stochastic, a deterministic, and a hybrid one (combining the characteristics of both). They are modeled in different levels of abstraction.

The access points are connected to the control server via a wired backbone network. The routes, over which packets are transported, are defined before the simulation. The control server stores the routes for each end-to-end connection and statically routes the packages.

Each sensor node has an associated energy consumption function in the model, which calculates the consumption according to the current activity of the sensor node. Based on the energy consumption function and the lifetime model, the energy-induced reliability (lifetime) of the system is determined during the simulation.

The driving force of our simulation is the application, which is described by an application specification. It contains application restrictions such as maximum end-to-end delay, maximum package loss etc. To give the applications individual properties, the generated traffic can be fed into each sensor node individually. Each application may have different traffic patterns for each phase in the schedule. At the moment the generator can produce stochastic, periodic, and single-packet data traffic.

III. Implementation

To implement the model we used a framework supporting an iterative agile development process [10].

The model was implemented in the MLDesigner tool [11], [12], a multi-domain simulator. It allows the modeling of hierarchical models by using block diagrams. For our model we used the discrete event domain (DE) for the network as well as the lifetime model, and finite state machines for the energy consumption. To achieve a high scalability of a model (i.e., independence of the model size and structure from the number of similar objects in the system), MLDesigner allows generating instances of a template during runtime of the simulation (termed as *dynamic instances*). It is not only a network simulator like NS-2 [7] or OMNETT++ [8], more generic and simplifies system environment modeling known as the *Mission Level Design* approach. Libraries of modeling elements for a variety of application areas simplify modeling of complex systems [13]. It has been applied to ZigBee networks earlier [14], among many other application fields.

Components of the network are modeled as such templates and can be instantiated and configured during simulation time. The configuration of each component is provided by a database, which can be populated with data by an external configuration tool.

The instances are connected with each other by an event channel (medium). It calculates the error rate for each connection and detects collisions. In addition to that, we can add interferers to disturb the communication and capture channel failures. Performance parameters and errors are saved in a database during simulation and can be analyzed afterwards.

IV. Example network

To validate our model, we create an example with a small network running two applications. In the next sections we will describe the structure of the actual network, the environment, and its applications. The network will be analyzed with three different scenarios afterwards.

A. Structure

The example model is based on the previously described model architecture. It contains a reduced amount of components to validate the accuracy of the model. In Figure 1 the logical structure is shown. For our simplified model we use four dynamic instances for access points (AP) and 30 dynamic instances for sensor nodes (SN). The four APs are connected to the control server via a backbone network with a daisy chain topology. The delay of data because of the daisy chain depends on the position of the AP in the chain. This lead to a additional delay on the backbone network for end-to-end transmissions.

For our network we use a hybrid protocol which provides deterministic and stochastic medium access. Figure 2 sketches the structure of an access point frame in the superframe of the hybrid protocol.

A frame in the protocol begins with a beacon, which is a start signal for the frame, followed by a beacon guard to reduce the chance of collision. Then the deterministic part starts. Every sensor node which is assigned to the deterministic part

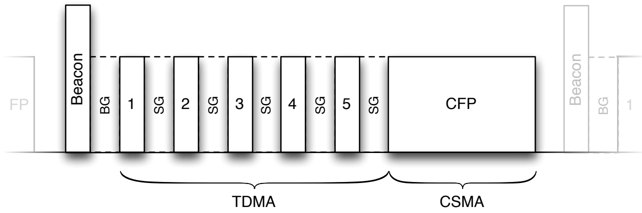

Fig. 2. Network protocol: Access point frame structure

has its own uplink and downlink slot for bidirectional communication. The slots are separated by guards. The stochastic part begins after the last slot guard. Sensor nodes assigned to this part access the medium via CSMA/CA. Table I shows the required parameters for the MAC layer in our example.

TABLE I
PARAMETER FOR THE MAC LAYER

parameter	value
superframe length	16000 byte
ap frame length	4000 byte
beacon length	2 byte
beaconguard length	2 byte
slot length	40 byte
slot guard	40 byte
slot count	40
csma length	400 byte
csma access mean	0.00256 seconds
csma wait retry	0.00256 seconds
byte time	0.000032 seconds/byte

B. Environment

The environment controls traffic generation and energy production models of the sensor nodes. For our example simulation we use one flight of an aircraft. The schedule can have 12 different phases.

TABLE II
THIS TABLE SHOWS THE SCHEDULE FOR A FLIGHT OF 30 MINUTES WITH DIFFERENT PHASES. THE WHOLE SCHEDULE TAKES ABOUT 4000 SEC.

description	phase	starttime
aircraft at gate	parking	1 sec
roll to runway	taxi	901 sec
takeoff	takeoff	1081 sec
climb	climb	1231 sec
cruise	cruise	1471 sec
descend	descend	3271 sec
landing	landing	3600 sec
roll to gate	taxi	3700 sec
park at gate	parking	3811 sec

In our example the sensor nodes are powered by a battery and they will not recharge by energy harvesting. A future extension will be model part describing harvesting based on operational phases and sensor locations.

C. Applications

For our example we used two applications with different traffic characteristics. A detailed specification for both applications including end-to-end delay requirements is shown in Table III.

TABLE III
SPECIFICATION OF APPLICATION PARAMETERS

	Application 1	Application 2
characteristics	periodic	periodic
time critical	no	yes
deadline	-	1 sec
number of SN	10	20

The first application is a non-restricted one, which produces different loads of data traffic in different phases. It serves as a background load for our simulation. The second application captures the behavior of a typical logging application, which measures the temperature of a sensor in a defined interval. The application has an additional critical aspect relating to the end-to-end delay and the number of lost packages. Table IV describes traffic patterns for the particular flight phases.

TABLE IV
EXAMPLE TRAFFIC PATTERN FOR APPLICATION 2

phase	characteristic	frequency	data size
parking	periodic	10 sec	36
taxi	periodic	5 sec	36
takeoff	periodic	1 sec	36
climb	periodic	1 sec	36
cruise	periodic	10 sec	36
descend	periodic	2 sec	36
landing	periodic	1 sec	36

There are critical phases in which significantly more data has to be acquired and transferred. Typically, this may happen during takeoff and landing. The length of one packet is the same, however, because only the sample frequency and not the amount of data per sample is changed.

D. Scenario

To analyze the impact of the protocol on timing requirements, energy consumption, and lifetime, three different experiments are considered (c.f. Table V).

TABLE V
SCENARIO AND CORRESPONDING PROTOCOL FOR EACH APPLICATION

scenario	application 1	application 2
1	CSMA	CSMA
2	TDMA	CSMA
3	TDMA	TDMA

In the first scenario, both applications access the medium via CSMA. This means, that they can block the other application while sending. The second scenario uses a hybrid medium access protocol, in which one application uses TDMA and the other CSMA. Finally, both applications are communicating via TDMA in the third setup.

E. Results

In the following section the results of our simulation in the three scenarios are described. Figure 3 shows the development of package delay of both applications during simulation for the first scenario. The figure shows the mean delay of application 1 (green line), application 2 (blue line), and overall (red line). The horizontal line shows the maximal delay requirement of application 2.

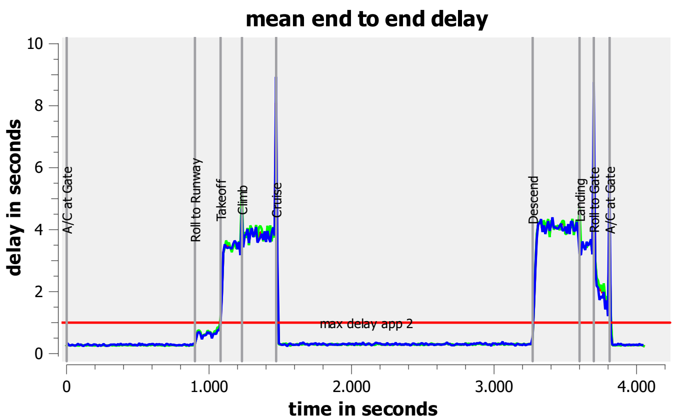

Fig. 3. Mean delays per application

As a result, both applications run in the same part of the protocol and influence each other while trying to access the medium with a concurrent protocol. The mean delay corresponds to the current load of the network. Application 1 — with no restriction — can run without any problems.

Unlike application 1, application 2 has a time restriction for its maximum end-to-end delay (represented by the red line). The delay fails to meet the requirement five times (takeoff, climb, descend, landing) when the load is high. The peak delay is about nine times higher than the limit. With these results it is obvious that running both applications with CSMA will not fulfill their requirements.

In scenario 2, the TDMA part of the protocol is used for the time-critical communication. Application 1 stays in the CSMA part. The resulting behavior of the mean delay is shown in Figure 4. The figure shows the mean delay of the second scenario for application 1 (green line), application 2 (blue line), and overall (red line). The horizontal line is the maximm delay requirement of application 2.

With the results it is easy to see that the delays of both applications are now independent. They do not influence each other unlike in the previous scenario. The mean delay of application 1 is still high in phases with high traffic, but the peak delay is much smaller than in the previous scenario. The reason is that the number of sensor nodes in the CSMA part drops from 30 to 10. Thus only ten sensor nodes are accessing the medium concurrently.

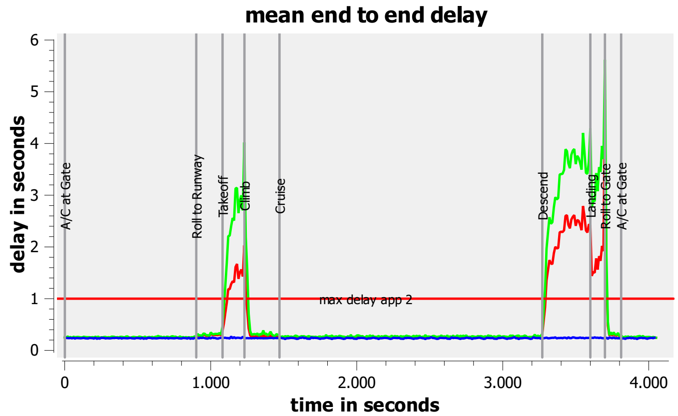

Fig. 4. Per-application delays, second scenario

The other 20 sensor nodes only have one time slot in the TDMA part of the protocol. The maximum delay no longer corresponds to the network load but to the length of the super frame and data generation of the sensor node. In our model, the super frame amounts to 0.5 seconds. For redundancy reasons, each sensor node can "‘see"’ at least two access points. Each sensor node assigned to the TDMA part can thus send a package two times in a super frame. As expected, the mean delay of application 2 is around 0.25 seconds. Thus the model-based simulation analysis shows that the requirements can be fulfilled with a hybrid distribution of the applications.

In the last scenario we assign both applications to the CSMA part of the protocol. The results are depicted in Figure 5. It shows the mean delay for application 1 (green line), application 2 (blue line), and overall (red line). The horizontal line is the maximum delay requirement of application 2.

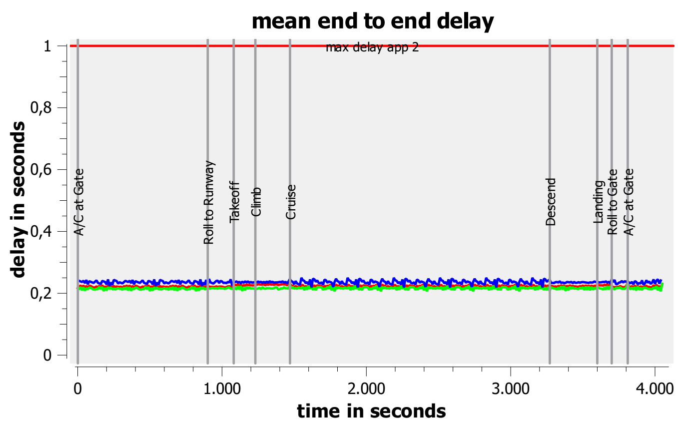

Fig. 5. Delays for scenario 3

Now each of the 30 sensor nodes communicates in its own time slot, and will not influence each other anymore. As in scenario 2, the mean delays of the applications are only influenced by super frame length and data generation of each node.

It is important to choose the parameters of the MAC layer depending on the requirements of applications. For example, a critical application, which needs a value every 0.5 seconds, cannot run with a MAC with a super frame length of 2 seconds.

As a result we can check requirements of the applications with a model-based approach. By simulating an application in its different phases, the model delivers more accurate results than a simplifying worst-case simulation. Transient effects at mode changes can be analyzed. As an example result to be demonstrated, in cases of very low data traffic a stochastic protocol can be efficient regard to energy consumption and latency. On the other hand, in a scenario with very high traffic and lots of sensor nodes, a deterministic protocol is more efficient.

V. Conclusion

The paper presents a wireless sensor network model for the design in a specific avionic environment. Quality of service parameters of an application can be evaluated more accurately: operational phases and their influence on system operation are specifically addressed. Simulation shows transient effects, for instance after mode changes. The application example shows how the model evaluation is used to check if MAC parameters are chosen well or if overload situations may occur.

Future work will be to predict lifetime by adding an energy map to the environment and abstracted energy harvesting components to the sensor nodes.

References

[1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. John Wiley & Sons, 2009.

[2] J. Beutel, "Design and deployment of wireless networked embedded systems," Ph.D. dissertation, ETH Zürich, 2005, shaker Verlag, ISBN 3-8322-4760-2.

[3] Y. Liu, W. Zhang, and K. Akkaya, "Static worst-case energy and lifetime estimation of wireless sensor networks," *JCSE*, vol. 4, no. 2, pp. 128–152, 2010.

[4] D. Chen and P. K. Varshney, "QoS Support in Wireless Sensor Networks: A Survey," in *Proc. of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA*, Jun. 2004.

[5] S. Tschirner, L. Xuedong, and W. Yi, "Model-based validation of qos properties of biomedical sensor networks," in *Proceedings of the 8th ACM international conference on Embedded software*, ser. EMSOFT '08. New York, NY, USA: ACM, 2008, pp. 69–78. [Online]. Available: http://doi.acm.org/10.1145/1450058.1450069

[6] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi, "Uppaal - a tool suite for automatic verification of real-time systems," 1996.

[7] K. Fall and K. Varadhan, *The ns Manual (formerly ns Notes and Documentation)*, Jan. 2009. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[8] A. Varga and D. History, "Omnet++ discrete event simulation system, version 3.1 edition," 2005.

[9] P. A. Farrington, H. B. Nembhard, D. T. Sturrock, G. W. Evans, and X. Chang, "Network simulations with opnet."

[10] R. Maschotta, S. Jäger, T. Jungebloud, and A. Zimmermann, "A framework for agile development of simulation-based system design tools," in *Proc. IEEE International Systems Conference (SYSCON'13)*, 2013.

[11] G. Schorcht, P. Unger, A. George, I. Troxel, D. Zinn, H. Salzwedel, K. Farhangian, and C. Mick, "System-level simulation modeling with MLDesigner," in *11th ACM / IEEE Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2003.

[12] A. Agarwal, C.-D. Iskander, R. Shankar, and G. Hamza-Lup, "System-level modeling environment: MLDesigner," in *IEEE Int. Systems Conf. (SysCon 2008)*, Montreal, Canada, April 2008.

[13] T. Baumann, A. Pacholik, and H. Salzwedel, "Performance exploration with MLDesigner using standardised communication interfaces," in *Proceedings of the DATE '07*, 2007.

[14] F. Lohse and V. Zerbe, "Model based performance estimation in zigbee based wireless sensor networks." in *27th. IEEE International Conference on Microelectronics (MIEL 2010)*, 2010, pp. 307–310.