Challenges of Simulating Robust Wireless Sensor Network Applications in Building Automation Environments

Michael Galetzka, Jürgen Haufe, Michael Lindig, Uwe Eichler, Peter Schneider
Fraunhofer Institute for Integrated Circuits (IIS)
Design Automation Division
Zeunerstraße 38, 01069 Dresden, Germany
Michael.Galetzka@eas.iis.fraunhofer.de

Abstract

Wireless sensor networks (WSN) get increasing importance in different application areas. In building automation, applications on WSN are expected to utilize a large number of nodes in a heterogeneous environment. They are too complex to be developed in the traditional way of prototyping and debugging. In fact, new techniques are needed to support the design of WSN applications efficiently. Simulation is one important method to decrease design time and to increase design quality. This is crucial for applications in building automation because very high reliability and robustness are required.

In this paper major challenges of simulating wireless sensor network applications are discussed. Using a WSN simulation framework, functional simulation and the evaluation of robustness of WSN applications in building automation environments are demonstrated.

1. Introduction

Currently, the market for wireless sensor networks (WSN) is growing rapidly in several application areas. Hence, a growing number of academic publications deals with evaluation of wireless networks including simulation of network topologies, routing strategies and other networking issues, specially focusing on WSN now. This includes modeling and evaluating network performance parameters as well as energy consumption.

Building automation is dominated by control tasks e.g. for lighting, heating and air condition. Therefore, using wireless sensor networks requires to consider the specific demands of distributed control and to ensure a high robustness of the entire system.

Another characteristic of WSN in this application area is the large number of nodes forming one network. Even simple applications with light and shutter switches and a few temperature sensors per room will accumulate in some hundred rooms to several thousands of nodes within the whole building. If this infrastructure will be used and extended for distributed control applications, e.g. air conditioning, the complexity of such a system will be enormous. It will not be feasible to develop and implement those networks and applications in the traditional way of prototyping and debugging. Rather, there has to be provided simulation based design support throughout all stages of the development process. This requires a modeling strategy and environment which covers several levels of abstraction, from pure functional models to detailed models featuring specific network stacks and precise channel behavior. On the other hand, the specifics of a dedicated building environment have to be considered as well – in a certain way at all levels of abstraction. The resulting challenges go beyond the usual network simulation tasks and need a thorough analysis in order to manage the trade-off between mastering the complexity and, consequently, the need for a significant speed-up on one hand and the necessary accuracy of the simulation on the other hand.

Section 2 discusses the characteristics and constraints for wireless sensor networks in building automation. In section 3 a definition of the term robustness will be developed with respect to these characteristics and constraints. Then, section 4 describes challenges of modeling WSN applications and their robustness in building environments and discusses approaches to overcome them, e.g. the handling of the wireless channel and its disturbances and interferences.

2. Characteristics and constraints

In order to identify and name characteristics and constraints pertinent to WSN applications in building automation, two topics offering different perspectives shall be considered: Sensor-actuator systems and wireless or mobile ad hoc networks (MANETs).

2.1. Sensor-actuator systems

A simple sensor-actuator system is, for example, a thermostat for a radiator. It roughly consists of a temperature sensor, an actuator for the valve and a control unit. From a communication view the sensor acts as a data source, the actuator acts as a data sink and the control unit does some kind of data processing. If we now locate
the single elements of such a sensor-actuator system somewhere distributed in one or several room(s), we will have to transmit data over a communication network. Thus, a WSN application can be seen as a spatially distributed sensor-actuator system utilizing a communication network for sending data from the sensor(s) via an open-loop or closed-loop control application to the actuator(s) (see also Figure 3).

2.2. Wireless ad hoc networks

Wireless or mobile ad hoc networks (MANETs) have been studied since the mid 1990s. The most important focus of research activities on this area has been routing mechanisms and protocols. Due to their ad hoc nature, wireless ad hoc networks do not have an infrastructure. Rather, the network topology is self-configuring. That also means, each node needs to have some minimal routing information to communicate with the other network nodes. The mobility of nodes causes a more or less frequent change of the network topology.

Since all nodes of one MANET share the same communication medium anyway, they potentially can communicate with all nodes within their radio range. Thus, MANETs often use mesh topologies instead of tree or clustered tree topologies. This makes them more fault-tolerant against (temporary) outages of single routing nodes. In turn, it raises the question how to take care of energy-efficiency of the routing mechanisms. The more nodes are able to route packets the more fault-tolerant the network becomes because of the mesh topology. Routing ability on the other hand affects the energy-efficiency since nodes may not remain in sleep mode at times when they are not used by the application. [11; 17]

2.3. Specifics of WSN in building automation

With the considerations above a wireless sensor network can be seen as a spatially distributed sensor-actuator system which utilizes an ad hoc network for data transmission. This also implies possible issues because of collisions due to the shared communication medium. And of course, there are the common potential connectivity problems as a consequence of the general wireless nature of the communication [11].

Specifics of WSN in building automation environments are the large number of miniaturized nodes (as motivated in section 1), which is expected to grow continuously as the applications become more pervasive. This implies not only the need for very low costs per node, but also hard restrictions with respect to the energy consumption. WSN nodes in building automation must be able to get their energy from batteries which will never have to be changed during the lifetime of the nodes or must be completely energy autarkic using energy harvesting technologies.

Therefore, WSN nodes are resource constrained and have to use transmission protocols which work energy efficient itself. Among others this means relatively small data rates (250 kbit/s in case of ZigBee) and low-complexity communication protocol stacks. On the other hand only little or moderate mobility of a few nodes in the network has to be dealt with in this application area.

An interesting challenge arises for building automation applications: The more complex the network itself gets the more it would become sensible to use the same network infrastructure by different automation tasks or applications. With the traditional OSI layered view of networks this wouldn’t be a problem. But recalling the limited resources of the network and the nodes itself, this will be the most challenging design issue for building automation applications. These applications will compete for the resources in terms of energy, data rate and processing power. The different OSI layers cannot be considered as completely independent. Rather, the impact of the very nature of the applications has to be considered in a cross-layer view already during the design process of an automation system [11].

Further relating aspects cover the optimization of the communication by data aggregation within the network, the radio planning within buildings (there is no traditional radio planning since the sensor/actuator nodes have to be placed according to their function), security and privacy, and the handling of interferences and other influences affecting the network function in general (walls, people, furnishing etc., see also section 3.4).

2.4. Technical realization of nodes

Currently, the most widely used communication standard for WSN is IEEE 802.15.4 [5]. Like all IEEE 802.x standards it covers only the physical transmission and the medium access layer. On top of this standard, a number of networking standards like ZigBee [22] or WirelessHART [20] are available. Other communication protocols are under investigation or in use for WSN, e.g. IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [6] or various proprietary protocols.

Regardless of the actual communication protocol used and regardless of the future development within this
topic, the specifics of WSN as discussed in the sections above will remain valid in general. Thus, without loss of generality we assume IEEE 802.15.4/ZigBee networks for the following discussions.

As shown in Figure 1, a WSN node generally consists of the sensor(s) or actuator(s), a radio transmitter chip with antenna, and a microcontroller. Most of the communication protocol stack will usually be implemented in software and runs together with the actual application(s) on the microcontroller. Depending on the application to be implemented on such a node, the microcontroller will typically be an 8bit controller with limited resources (e.g. 8 kB RAM, 128 kB FLASH operating at 4 or 8 MHz clock frequency).

3. Evaluation of robustness

The term robustness is used in a broad meaning in various contexts also in technical systems. Generally, robustness describes the ability of a system to function correctly even if conditions arise inside or outside the system which do not match the normal operation conditions. In this section, we will develop a definition of the term robustness in the context of WSN in building automation.

3.1. QoS from network view

First, we will discuss the term quality of service (QoS) which is closely related to the term robustness in a network. A service in telecommunications is defined as the ability of a network to transmit dedicated information [14]. There exist a variety of different definitions of QoS. One common definition in [7] describes QoS as “the collective effect of service performance which determines the degree of satisfaction of a user of the service”.

QoS requirements at different communication layers can be characterized by dedicated QoS parameters at each of these layers. Each network layer has its own utilization of QoS parameters (see Figure 2 for some examples). These technical parameters can be mapped across the layers from PHY to application layer, and finally to those categories of QoS which a user of an application perceives and which hence focus on certain user-perceived effects (rather than on their causes within the network), even though the QoS parameters at each network layer contribute to these effects. [3]

![Figure 2. QoS parameters at different network layers](image)

3.2. Distributed applications

From the discussion in section 2 follows that we have to distinguish between the applications on a dedicated WSN node (the application layer) and the distributed application(s) in a WSN which form the sensor-actuator system(s) to be investigated.

Assume a (closed-loop) control application as shown in principle in Figure 3. Such a control application can be considered and simulated as a model of a certain sensor-actuator system. When we further assume now that parts of the application might be distributed to different nodes of a WSN as discussed in section 2.1, in the first instance we have to take into account the delay between those parts caused by the transmission of data over the network (introduced as delay elements in the

![Figure 3. Distributed application model mapping into a WSN](image)
model in Figure 3). In the further course, the control application will actually be divided into parts which will be mapped and implemented each as an application layer on a WSN node. These node applications communicate with each other and therefore virtually form the original distributed control application.

3.3. QoS from application/user view

Following the definitions in section 3.1 and considering the discussion in section 3.2 we now can specify what QoS from application or, moreover, user perspective means: The application simply shall function as specified or expected. In particular, e.g. if the light switch is pressed, the (correct) lamp should illuminate; the climate control should make sure that the right temperature and air humidity will be held over the day etc.

Obviously, this requires that at different network layers data packets are transmitted correctly and timely. Depending on the application, the QoS category timely can require different latency times at network layer for different applications. Furthermore, this may imply certain bit error rates at PHY and subsequently certain packet error rates at MAC layer, since due to the retransmission scheme within the IEEE 802.15.4 MAC layer the packet error rate partly translates into the latency/delay at NWK layer (see Figure 2).

3.4. Robustness

The definition of robustness of the WSN application follows the same idea as the QoS discussion in the previous section. The “ability of a system to function correctly”, as formulated at the beginning of this section, translates to a certain quality of service from application/user view. Thus, the definition of “function correctly” may vary to a certain degree depending on the application itself.

However, there may occur “conditions inside or outside the system which do not match the normal operation conditions” and which may affect the required QoS. The robustness of a WSN application is the ability of that application to operate within the required QoS parameters of the distributed application even if the WSN does not work in the specified behavior in all parts of the network.

In a building automation environment, there are a number of possible reasons for operating conditions deviating from the normal state:

- outage of single nodes e.g. due to empty batteries or insufficient energy-harvesting conditions,
- minor (possibly temporary) changes in the environment affecting the signal reception conditions of certain nodes,
- people moving around influencing the attenuation of the wireless channel
- interfering radio signals from the normal building infrastructure (e.g. Wi-Fi access points with adaptive power)
- varying load situations within the network due to extraordinary events
- unreliability of sensor data because of physical restrictions of the sensors (extreme environmental conditions)

Of course, this enumeration may be further completed for special application environments.

There is a close relation between the terms robustness and fault-tolerance. The difference is that in a wireless network the term fault (e.g. the outage of a node because of empty batteries) cannot clearly be distinguished from transmission effects and uncertainties inherent to a wireless communication system.

Simulating the robustness of a WSN application requires two prerequisites:

- a validated simulation model reflecting the “normal” operation of the WSN,
- models for variation of the conditions described above

After that, a number of simulations have to be performed varying the conditions relevant for robustness and testing that the application level QoS parameters stay within the application-specific boundaries.

4. Challenges in simulation

This section discusses challenges in simulation of WSN applications especially in building automation environments. It describes approaches to model WSN applications for simulating their robustness as defined in section 3.4. It shows some practical experiments and results with regard to the handling of the wireless channel and its disturbances and interferences.

4.1. Aims of modeling and simulation

The aims of the simulation follow from the discussions and definitions in section 2 and 3:

- Evaluate the reliability and application-specific robustness of an entire WSN application and of measures to improve them.
- Perform a holistic simulation (application(s), sensor/actuator behavior, communication, and environment).
- Handle the trade-off between mastering the complexity and, consequentially, the need for a significant speed-up on the one-hand side and the necessary accuracy of the simulation on the other hand.

In addition to these general modeling and simulation tasks, evaluation of a WSN application may cover further aspects, like modeling and tracking the energy consumption of nodes or the degree of utilization of other resources as processor load and main memory [18, 19].
4.2. Simulation environment

We use the discrete event simulation environment OMNeT++ for our WSN design support framework. OMNeT++ is a general public-source simulator not dedicated to special application areas, but with a strong community in queuing, wireless, and ad hoc network simulation [16; 12].

Apart from providing the actual functional models, a number of general tasks have to be solved for modeling and simulation of wireless ad hoc networks: e.g. modeling the actual connections on the shared medium, generic handling of the channel and the communication layers, dealing with spatial location and mobility of nodes. On top of the simulator OMNeT++ there exists, among others, the wireless simulation framework MiXiM [8; 10], which provides base classes, models and mechanisms to handle these issues.

This base framework has to be extended in order to achieve the aims defined in the previous section. Thus, a simulation and modeling framework is developed which can be used to simulate WSN applications and evaluate their robustness characteristics. The following section shows three subtasks within this development with special challenges for building automation environments and reviews some preliminary results.

4.3. Channel modeling

As already mentioned, a wireless network shares a common transmission medium – the channel. Generally, all nodes within the network use the same channel, thus, physically each node is connected to each other node (independently from the logical network structure as depicted in Figure 4).

However, which nodes actually may receive signals from other nodes depends on the radio coverage of each node which again will be influenced by the characteristics of the channel. This radio coverage or personal operating space (POS) is indicated in Figure 4 as circles for the three middle nodes. In the simulation framework based on MiXiM there is a double-stage handling of these connections:

First, all possible physical connections are maintained by a module called connection manager. It uses a very simple channel model which only depends on the transmission power and the distance between two nodes as a worst case. The path loss $L$ increases with the power $\alpha$ of the distance $d$ as shown in the following known formula:

$$ L = L_0 + 10 \cdot \alpha \cdot \log(d) \ [dB] \quad (1) $$

Second, if a certain packet is actually transmitted, it will be received by all nodes connected with each other by the connection manager in stage 1 described above. Then, each node determines the received power and the quality of the signal (e.g. signal to noise ratio, bit error rate) by evaluating a more or less dedicated channel model.

There are several effects which have to be taken into account in a building environment.

**Multi-path reception:** A node does not only receive a signal on the direct path (if at all). Rather, the signal may be reflected and diffracted on walls, doors, furniture, and other items and reach the receiving node on various indirect ways with different signal strengths, but due to interactions with obstacles and the transmission delay also with different phases and therefore increasing or decreasing the resulting received signal strength. This multi-path transmission may be simulated by so-called ray-tracing simulations. Ray-tracing based tools [4], as they are typically used for propagation modeling and radio planning for mobile or WiFi networks, are very powerful for those purposes, but are too time-consuming for simulation of WSN applications with thousands of nodes.

Figure 4. WSN nodes and POS in a building environment

Figure 5. Drops in signal reception due to multipath propagation

Figure 5 shows the signal strength measured in an office environment depending on the distance (without any obstacles in between). Due to the multi-path transmission occasional drops in signal reception of up to 20 dB can be observed. That means, if in a WSN a sensor node will be placed a few inches next to the planned position there can be a significant degradation of the received signal quality. Since ray-tracing cannot be used, this effect has to be taken into account within a simulation by varying the channel model parameters statistically for sensitive nodes and checking the functionality at net-
work and application level (a kind of monte-carlo simulation).

**Disturbances and interferences:** In a building environment there are several effects influencing the signal transmission. One of these effects is the simple fact that people are walking around and possibly affecting the channel behavior. Figure 6 shows the distribution of the received signal strength (energy detection level – ED) of 5000 packets sent with 3 dBm and received in a floor at 12 meters distance over a period of about 155 seconds with two people randomly but continuously walking around between the nodes. Reference measurements show that without people normally the ED should only vary between about three adjacent values.

![Figure 6. Distribution of ED with and without people randomly but continuously walking round the nodes](image)

This effect has to be considered in the channel model by statistical means again. But other than statistically varying parameters over different runs (monte-carlo simulation) as for the multi-path reception effects, the channel parameters now statistically vary over the simulated time within each run. Similar considerations are valid for interferences by Wi-Fi access points and terminals or Bluetooth enabled mobile phones.

**Obstacles:** The simple path-loss channel model mentioned above does not provide sufficiently accurate results when simulating wireless networks in practical environments. Therefore, a number of channel models are used for different application scenarios. Common channel models for mobile transmission are, e.g. Ricean or Rayleigh channels. In office like environments, we channel models for mobile transmission are, e.g. Ricean or Rayleigh channels. In office like environments, we have little or no fading effects because of limited mobility. Rather, the signal is attenuated by obstacles, mainly the walls inside the building. That’s why multi-wall models like the following are used:

\[ L = L_0 + 10 \cdot \alpha \cdot \log(d) + \sum_{i=1}^{k} kW_i \quad [dB] \]  

(2)

Here, the first part of the formula is the propagation model from (1). The second part is the multi-wall component, where \( k \) is the number of penetrated walls and \( W_i \) the attenuation of the wall of type \( i \).

These models are simple enough for efficient simulation of large networks, but reflect the circumstance that, beside the free-space part of propagation, the attenuation of the signal is mainly caused by the walls within the building. [2; 1; 21; 15]

**4.4. Modeling of obstacles**

The obstacles discussed above have to be dealt with inside the network simulation framework. In general, for each possible connection between two nodes the obstacles (resp. the walls between them) have to be identified. In [9], a simple obstacle manager (they call it object manager) is proposed for OMNeT++ based simulation frameworks. This object manager is conceptually also part of the MiXiM framework, but not implemented yet. In order to have a flexible but powerful mechanism to get the attenuation values for the multi-wall channel model, we decided to implement an obstacle manager in a slightly different way. We use a powerful optical ray-tracer, the open-source renderer PoVRay [13], which we integrated into the object manager of our simulation framework. The attenuation of the obstacles is modeled as a transparency for the light rays. An advantage of this solution is the potential to model even complex objects with describing also the structure of the surface and characteristics of the inside.

The ray-tracing based object manager supplies its connection related accumulated attenuation values to the connection manager each time a change in the spatial distribution of the nodes (and possibly the objects/obstacles) occurs. At a minimum, this has to be done once at the beginning of a simulation for a static network. If one node moves, the procedure has to be repeated for all affected nodes and their connections to the moving node. The PoVRay based object manager calculates the attenuation \( W_i \) and the distance \( d_{wi} \) the beam covers through the wall \( i \) according to the following formula (\( \beta \) is the fade power in the ray-tracer):

\[ W_i = 10 \cdot \log(1 + (d_{wi} / d_{ij})^\beta) \quad [dB] \]  

(3)

Subsequently, \( d_{wi} \) decreases the overall distance \( d \) from (2). The object manager is parameterized by determining the fade distance \( d_{fj} \) which is the distance the light has to travel through an object to be decreased to the half of its intensity. This value has to be determined by measuring the actual attenuation of the wall for a IEEE 802.15.4 signal:

\[ d_{fj} = d_{wi}(P_s / P_r - 1)^{1/\beta} \]  

(4)

where \( P_s \) and \( P_r \) is the sent and received power respectively and \( d_{wi} \) the distance between sender and receiver for this parameterization measurement.

As a first approximation we consider only the direct beam without reflections and diffractions. The model can easily be configured so that reflections up to a certain degree may be included too. There has to be further
investigation regarding the impact on the performance of the simulation with such extensions.

Figure 7. Optical ray-tracing for evaluating obstacles between connecting nodes

Figure 7 visualizes the optical ray-tracing for a setup of 10 nodes, rendered by the PoVRay module which is integrated into our simulation framework. Figure 8 shows the consequences for the physical connections. With the standard MiXiM connection manager, most of the nodes have a pairwise connection (except nodes on opposite edges of the playground, e.g. the upper left node[0] and the lower right node[5]). But with the extended connection/object manager the physical connections are reduced as shown in Figure 8.

Figure 8. Connections with consideration of ray-tracing for multi-wall model

4.5. Abstraction of network model

As a first step, actual network stacks are used as models for PHY and MAC [18] (and in the future also for NWK and APS layer). The advantage of this approach is that almost the same code can be used within the model and on the actual microcontroller implementing the real node (see Figure 1). Thus, this approach cannot only be used to evaluate the functionality of the whole system concept but also to test and debug the embedded software implementation.

The drawback is the complexity of the model itself, which not only requires some effort to port the software, but also downgrades the performance of the overall simulation. Therefore, the next step has to be the abstraction of the communication stack to save computing effort at the lower layers of the communication stack (PHY, MAC, partly NWK) by hiding some details of these layers while the essential functionality at application layer remains unchanged and usable for simulation. Figure 9 illustrates the principle of the method.

Figure 9. Abstraction of communication model

In general, the packets will not be sent all layers from application down to the PHY, transmitted over the physical channel, and then received vice versa all layers up at the receiver’s side. Rather, they will be transmitted e.g. on network layer over a virtual channel.

However, the connection conditions as discussed in sections 4.3 and 4.4 have to be emulated accordingly at this more abstract level. For example, the bit error rate $P_{BER,PHY}$ at PHY layer as it follows from the physical channel conditions will translate to certain latencies $l_{NWK}$ (affected by the retransmission mechanism of the MAC) and to a certain remaining packet error rate $P_{PER,NWK}$, as already introduced in section 3.1.

$$l_{NWK} = \begin{cases} l_0 & \text{with } 1 - P_{PER,MAC} \\ l_0 + l_g & \text{with } P_{PER,MAC}(1 - P_{PER,MAC}) \\ l_0 + 2l_g & \text{with } P_{PER,MAC}^2(1 - P_{PER,MAC}) \end{cases}$$

(5)

with $P_{PER,MAC} \approx b \cdot P_{BER,PHY}$, ($b$ number of bits per packet, $P_{BER,PHY} << 1$), $l_0$ the latency for transmission of one NWK packet, $l_g$ the additional latency caused by one MAC retransmission.

Another issue which has to be considered when developing abstract and time-efficient models at NWK layer is the clear channel assessment. While it can be simulated quite accurately at PHY/MAC layer, it has to be modeled statistically at NWK layer, since the latency is generally not the time the channel will be occupied by the sending node.

These and similar considerations are a matter of current and future work.
5. Conclusions

In this paper, we discussed some major challenges of simulating wireless sensor network applications in building automation environments and evaluating their robustness. We developed a terminology and methodology how to handle aspects of robustness in those specific cases. Eventually, we described practical implementations and results with regard to the handling of the wireless channel and its disturbances and interferences within our simulation framework.

Further activities will focus on the reduction of complexity by providing models at different abstraction levels (as discussed in section 4.5) and on combination of application level modeling (as shown in Figure 3) with the network simulation framework. This will be complemented by activities for a user-friendly interface, e.g. automatic import of 3D building models.

Acknowledgements

This work has been partially funded by the Saxon State Ministry for Economic Affairs and Labor and the European Union.

References


[22] ZigBee Standards Organization, ZigBee Specification 053474r17, Jan. 2008