

Beacon Frame based Network Simulation using TrueTime Network Simulator

(Poster Abstract)

Christian Roßberg, André Froß, Daniel Froß, Ulrich Heinkel
Chemnitz University of Technology
Faculty of Electrical Engineering and Information Technology
Professorship Circuit and Systems Design

{christian.rossberg, andre.fross, daniel.fross, ulrich.heinkel}@etit.tu-chemnitz.de

ABSTRACT

This paper presents an enhancement of a network simulator which allows the simulation of beacon-enabled IEEE 802.15.4 wireless networks. The wireless communication standard IEEE 802.15.4 implements two basic modes, the non-beacon-enabled and the beacon-enabled mode. For low power networks the beacon-enabled mode is most suitable. A time base is given and inactive periods are predefined by a superframe structure. Another advantage of the beacon-enabled mode is that in a cell the nodes can operate as real-time network. *TrueTime* supports a lot of wired and wireless protocols but the beacon-enabled mode of the IEEE 802.15.4 protocol is not implemented. Hence, the simulator was enhanced by the beacon-enabled mode.

Categories and Subject Descriptors

I.6 [Simulation and Modeling]: Miscellaneous

General Terms

Design

Keywords

IEEE 802.15.4, TrueTime, network simulation

1. INTRODUCTION

The interest in networked control systems has strongly increased in recent years. A low cost and low power approach are wireless networks based on the standard IEEE 802.15.4 [2]. In this paper we choose the *TrueTime* [5] network simulator and extend the functionality to support the beacon-enabled mode of IEEE 802.15.4. This mode allows more energy efficiency and increases the flexibility in wireless networks. But several degrees of freedom and a lot of functionality introduced by the mode lead to complex implementations. Hence, the beacon-enabled mode is often neglected.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTools 2010 March 15–19, Torremolinos, Malaga, Spain.
Copyright 2010 ICST, ISBN 78-963-9799-87-5.

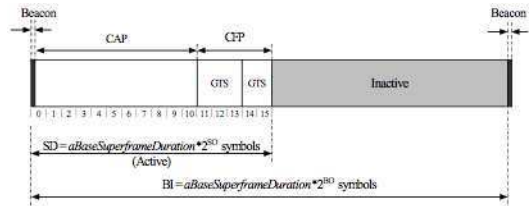


Figure 1: Superframe structure of the IEEE 802.15.4 protocol [2]

For adding this support we analyze the simulator functionality and add missing modules and functions. A programming interface allows the implementation of new functions and new blocks into MatLab/Simulink.

2. THE STANDARD IEEE 802.15.4

The standard IEEE 802.15.4 specifies the physical layer and medium access control of low-cost and low data rate WPANs (Wireless Personal Area Networks). Data rates up to 250 kb/s can be achieved in different frequency bands at 868 MHz, 915 MHz and 2.4 GHz. The CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism controls the channel access. Optionally, GTS (Guaranteed Time Slots) can be used for real-time operation. In this mode data transmissions are only allowed within the superframe structure, see figure 1. The superframe is divided into 16 equally sized slots, the first slot is used for the beacon. The master node sends this beacon signal cyclically for synchronizing all nodes of the network. The beacon also contains information about the format of the superframe. An additional inactive period is allowed after the 16 slots. The active period is divided into a contention access period (CAP), where nodes access the channel by using CSMA/CA, and a contention free period (CFP), where each slot is assigned to a single node.

3. THE TRUETIME SIMULATOR

TrueTime [1] is a network simulator for wired and wireless networks based on MatLab/Simulink and has been developed at the Lund University. Under several points of view the usage of the *TrueTime* simulator is favorable because it offers the advantage of co-simulation facilities. The *Network-* or *Wireless Network-Block* represents the commu-

nication channel and the *Kernel Blocks* represent the network nodes. A kernel block is a processing unit with generic interfaces. Main functions of the kernel and network are written in C++. Higher network layers are not implemented but can be realized by user code. The user code can be written either in MatLab m-Files or C++. The Application Programming Interface (API) is formed by a set functions which are necessary for the initialization and managing the behavior of the network and network nodes.

4. TRUETIME WITH GTS

As *TrueTime* supports only the non-beacon-enabled mode of IEEE 802.15.4, the API has to be adapted. At first the superframe structure was realized. Therefore, variables for storing the GTS parameters were added. The function *ttset-networkparameter* was extended to set the GTS parameters. After the implementation of the superframe structure a function for the generation and sending of beacon frames was necessary. The beacon is generated automatically by the wireless network function *ttwnetwork*. The main function is called by *runNetwork*. Within this function the sending routine of each wireless network protocol is executed. For the IEEE 802.15.4 with GTS the function is divided into three parts. The first part executes the beacon send command. The second part sends data within the CAP. In the last part all data of the CFP are sent. The active part is indicated by the slotcount and the GTS parameters. In the new file `message_types.h` the beacon frame format is specified. Several parameters can be set, but for the first implementation only the GTS parameters are interesting. So GTS nodes, GTS startslots and GTS endslots are assigned to the specified beacon frame fields. After the beacon the CAP starts. The function for sending data during the CAP changed from special functionality in [4] to common functionality. A parameter *batterylifeextension* was introduced, for more information about this parameter see the IEEE 802.15.4 standard [2]. The send function of the CFP was also modified. In `wnetwork_callback.m` the case “Zigbee with GTS” was added. This file is executed by the wireless network block and values in the blockmask will be overridden by default settings for different network types. The block-library `truetime.mdl` was extended by the *Remaining Block* which is a S-Function block with the corresponding S-Function *ttremaining*. The battery output is connected to the input of the *Remaining Block*. The block output is a vector of six variables which contain the calculated lifetime. The decision, whether a node operates as master or slave, is set in the *ttinitnetwork* routine. The function *ttgetkernelparameter* returns several kernel parameters like *energyconsumption*, *batterylifeextension* and so on. To get a network parameter the function *ttgetnetworkparameter* is used. For example returned values are *superframeduration*, *currentslot* and *gts_fields*. The most important new function is *ttsync-timeslot*. The network messages are stored in a queue. If the next hit is processed, all elements of the queue are sent. Therefore, it is important that the queue is filled already one superframeslot before the message shall be send. *ttsync-timeslot* fulfills this task, return values are the next segment to execute user code and the sleep time for the task. The next user segment is executed if the superframeslot reaches the send slot of the message. In figure 2 this principle of operation is shown. Problematically was the simulator extension for introducing the slot capabilities. Normally the

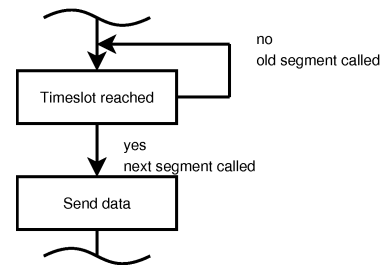


Figure 2: Principle of operation of *ttsync-timeslot*

simulator is acting after a message was put into the queue. For the IEEE 802.15.4 protocol the simulator will act when a slot is reached or a message is ready to send.

5. CONCLUSION

The presented extension of *TrueTime* allows the simulation of IEEE 802.15.4 beacon-enabled networks. The simulator is now usable for simulations of an existing network prototype at our professorship based on the IEEE 802.15.4a. IEEE 802.15.4a [3] defines new PHY layers for the network communication which allow additional distance measurements between the nodes. Therefore, algorithms for ranging, routing and autonomous network exploration will be implemented. Adding new protocols is very simple. This requires only some knowledge in C, C++ and MatLab. Transmission delays are not estimated and have not been regarded in calculations yet. In future work also the visualization performance should be improved.

6. ACKNOWLEDGEMENT

This work has been established in association to the Inno-Profile project “Generalized Platform for Sensor Data Processing” funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung) under project ID 03IP505.

7. REFERENCES

- [1] <http://www.control.lth.se/truetime>.
- [2] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Technical report, IEEE Computer Society, IEEE, 3 Park Avenue, New York, NY 10016-5997, USA, 2006.
- [3] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), Amendment 1: Add Alternate PHYs. Technical report, IEEE Computer Society, IEEE, 3 Park Avenue, New York, NY 10016-5997, USA, 2007.
- [4] N. Boughanmi, Y. Song, and E. Rondeau. Wireless Networked Control System using IEEE 802.15.4 with GTS. In *2nd Junior Researcher Workshop on Real-Time Computing - JRWRTC2008*, pages 21–25, 2008.
- [5] M. Ohlin, D. Henriksson, and A. Cervin. TRUETIME 1.5-Reference Manual. Technical report, Department of Automatic Control, Lund University, 2007.