Stochastic Simulation of Communications Networks and their Protocols

Project Description

Energy Consumption and Battery Modelling of Radio-Based Devices

(45 points)

Prof. Dr. Anna Förster
Dr.-Ing. Asanga Udugama
Dipl.-Ing. Jens Dede

Presentation and discussion: January 10, 2017 in class
Report and code submission deadline: January 17, 2017 via StudIP

Each submission should have two parts: a report describing the project and the OMNeT++ project files for the project in a Zip file. Please note that the format of your submission and its presentation quality is an essential part of the grading. Source code should also be commented properly. Group work of up to 3 students is encouraged.

General description. The general purpose of this project is to implement a sophisticated battery model in OMNeT++ for devices with a radio on board (e.g. Bluetooth, 802.15.4, etc.)

1. Select the model to implement Starting from the publication by Jongerden and Haverkort, "Battery Modeling", provided to you in class, select a model to implement. Consider following hints and rules:

- There is no ready-to-use differential equation solver for OMNeT++. However, you can use some other library for this.
- Only two groups are allowed to work on the same model. First come, first serve (by email directly to Jens Dede, jd@comnets.uni-bremen.de).

2. Implement the simulation model. Your model for each network node should consist of two individual modules: EnergyConsumer and BatteryModel. The minimum parameters for the EnergyConsumer are:

- Message length in bytes, set default to something meaningful.
- Sending interval, where 0 means that the node is not sending any messages, only receiving.
- Energy consumption (current) in Sleep mode

- Energy consumption (current) in Receive mode
- Energy consumption (current) in Send mode
- Sending frequency
- Minimum required voltage

Your concrete model might require additional parameters.

The EnergyConsumer model works as follows: it sends a message with the given message length every interval regularly, unless the interval is set to 0. In this case, the node does not send anything. After sending, the node goes to sleep and wakes up again in its next interval. If the node does not send anything, it wakes up every interval and stays in receive mode for a complete interval, then goes to sleep again. The first time it wakes up is a random number between 0 and interval.

Every time the node changes it's state, it signals this change to the BatteryModel with all relevant parameters (e.g. at least the new energy consumption level in the new state). Given this information, the BatteryModel calculates the current voltage of the battery. If the current voltage is lower than the minimum required voltage of the EnergyConsumer, the BatteryModel signals this to the EnergyConsumer which shuts down the node.

The parameters of the BatteryModel depend on the selected model to implement. At least the capacity of the battery is needed.

3. Simulation analysis We are interested in the node lifetime with various parameters. Select a transceiver from the given table for the basic setup for experimentation. Starting from this basic setup, explore the impact of the parameters sending interval (1 sec - 1 min, in steps of 5 secs), message length (64 to 1024 bytes in steps of 64) and three different scenarios for energy consumption (in the appendix). Plot the impact of the individual parameters on the node lifetime for the sender and the receiver in graphs. Pay attention to correct graph representation (axis titles, measure points connected or not, etc.)

Extensions (not mandatory, bonus points) You could extend your models. Here are some ideas:

- Consider different energy consumption levels depending on the sending power of the radio.
- Provide standard sets of parameter settings for some concrete radios (e.g. CC2420) and batteries.
- Port your model to INET
- Consider and model the energy consumption of switching between states
- Consider further power consumption elements on board of the devices: microcontroller, flash memory, etc.

Appendix: Sample Radio Power Consumption

	CC2420[a] 802.15.4 (Transceiver)	**CC2530**[b] 802.15.4 (Microcontroller & Transceiver)	**ESP8266**[c] WiFi (Microcontroller & Transceiver)	**CC2650**[d] BLE & 802.15.4 (Microcontroller & Transceiver)	**RFM69HCW**[e] 433MHz / 868MHz (Transceiver)	**RFD22301**[f] BLE (Microcontroller & Transceiver)
Min Voltage	2.1 V	2.0 V	1.7 V	1.8 V	1.8 V	2.1 V
Deep Sleep	20 µA	200 µA	10 µA	1 µA	0.1 µA	4 µA
RX	18.8 mA	24.3 mA	60 mA	6.1 mA	16 mA	12 mA
TX (min)	8.5 mA	28.7 mA	135 mA	6.1 mA	16 mA	12 mA
TX (max)	17.4 mA	33.5 mA	215 mA	9.1 mA	130 mA	12 mA

[a] http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=cc2420&fileType=pdf
[b] http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=cc2530&fileType=pdf
[c] https://www.adafruit.com/datasheets/ESP8266_Specifications_English.pdf
[d] http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=cc2650&fileType=pdf
[e] http://www.hoperf.com/upload/rf/RFM69HCW-V1.1.pdf
[f] http://www.rfduino.com/wp-content/uploads/2015/08/RFD22301.Data.Sheet.08.20.15_4.36PM.pdf