Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks

Voltage Scheduling of Peripheral Components on Wireless Sensor Nodes

Ulf Kulau[1], Stephan Friedrichs[2] and Lars Wolf[1]

[1]Technische Universität Braunschweig | IBR
{kulau|wolf}@ibr.cs.tu-bs.de | Phone +49 (0) 531 391-3290

[2]Max Planck Institute for Informatics | Germany
sfriedri@mpi-inf.mpg.de

DVS for Peripherals

While existing work focuses on the transceiver and the processing unit to increase the energy efficiency of wireless sensor nodes, it is missed that peripheral energy consumption may dominate that of the entire node.

Related to Dynamic Voltage Scaling (DVS), even peripherals' energy efficiency benefit from a downscaled voltage level, but different peripherals require different minimum voltage levels.

With this demo we combine theory and practice to present the implementation of an algorithm weighing off the benefits of a downscaled voltage level against the switching overhead, for calculating an optimal peripheral voltage schedule.

Voltage Scheduling Problem

Consider a sensor node with a set S of peripheral hardware devices. The energy consumption of each $s \in S$ is given by the following equation.

$$e_s(v) = \int_0^{t_s} I_s(v,t)\,d\,t$$

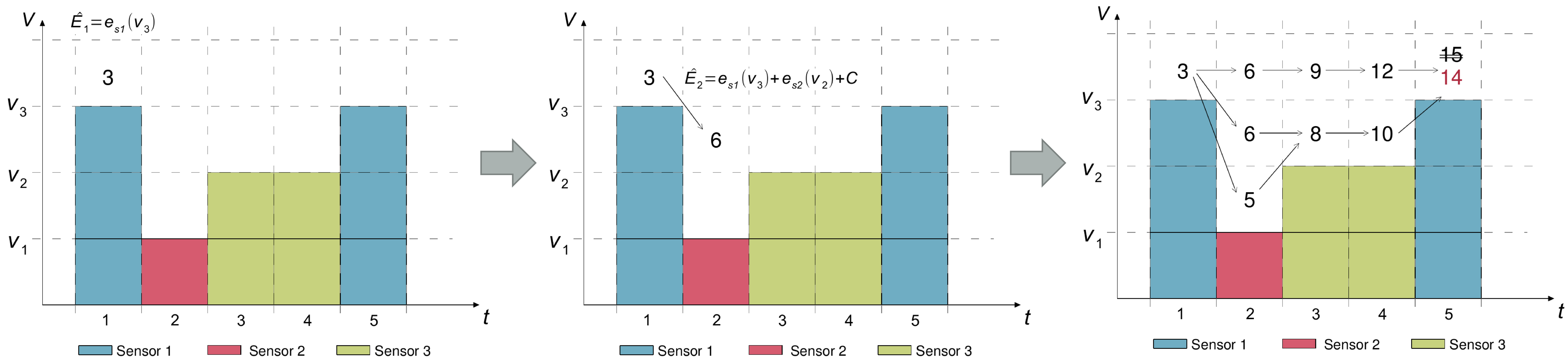
Each $s \in S$ has two attributes: 1. a minimum voltage $v_{min}(s)$ required to properly operate s, and 2. the total energy consumption $e_s(v)$ of all peripherals while only s is active, depending on the peripheral voltage v. For a constant amount C of energy, the switching overhead, the sensor node can adapt its peripheral voltage. The sensor node is presented a sequence of queries denoted by $[1, \ldots, n]$, so that the energy consumption E of a voltage schedule is given by:

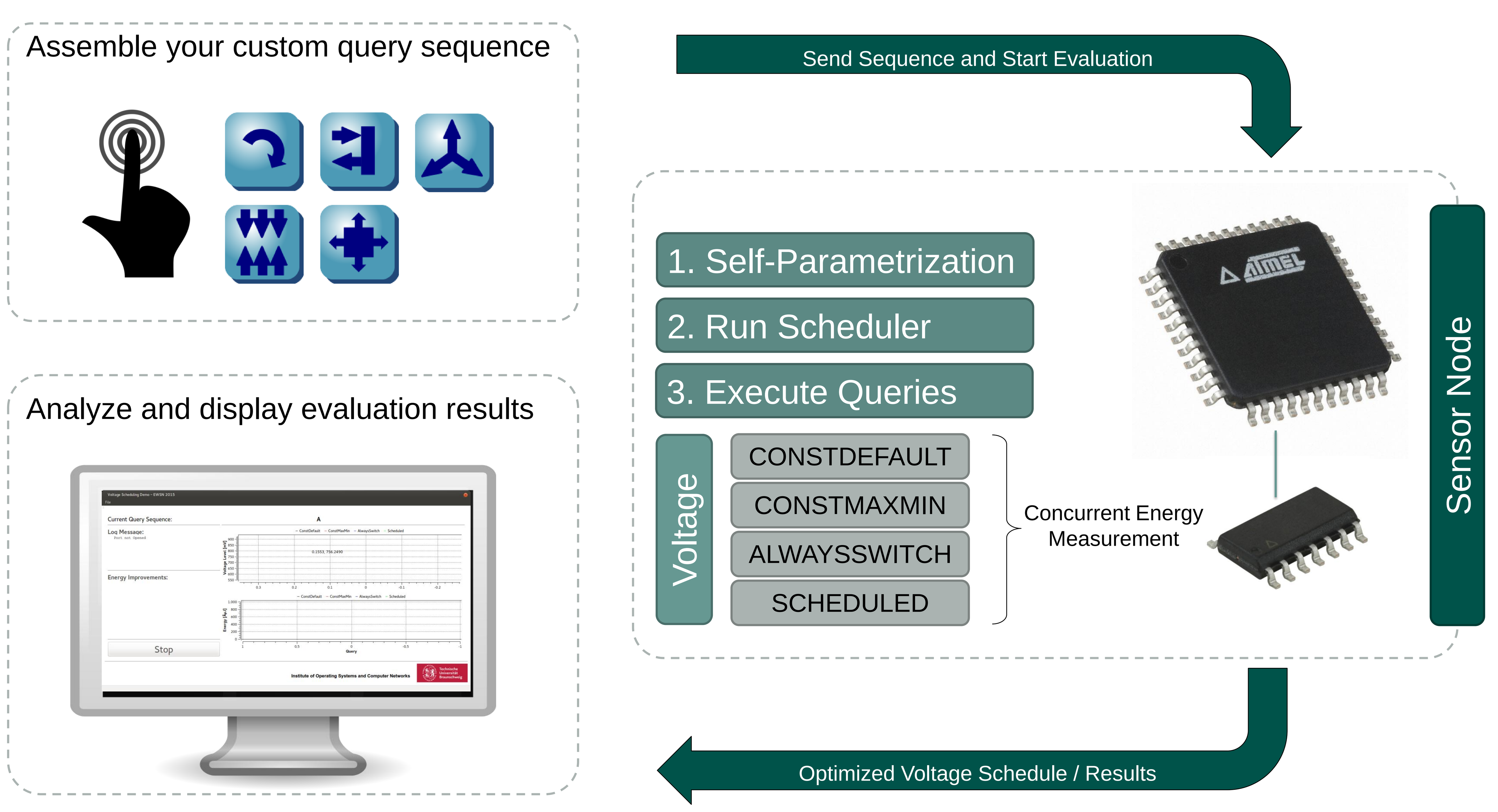
$$E = \sum_{i=1}^{n} e_{s_i}\big(v(i)\big) + \sum_{i=2}^{n} \begin{cases} C & \text{if } v(i-1) \neq v(i) \\ 0 & \text{otherwise} \end{cases}$$

Our goal is to minimize E, so we call a voltage schedule optimal if E is minimal

Scheduling Algorithm

We use dynamic programming to efficiently solve the recursion by determining E_i before E_{i+1}. The simplicity of this approach is sufficiently lightweight for typical WSN requirements like limited computational resources. The following example demonstrates the general functionality of the algorithm. Simplification: $C = 1, v_i = i$ and $e_{s_i} = v_i$





Demonstration

To show the benefit of voltage scheduling for peripherals, we demonstrate a live evaluation of custom query sequences.

A prototype HW executes the optimization algorithm to get the optimal voltage schedule for the given query. Finally the query is processed while the optimal schedule is compared against the trivial voltage strategies.

The second tiny MCU samples the current consumption of three trivial strategies *CONSTDEFAULT*, *CONSTMAXMIN*, *ALWAYSSWITCH* and of course the optimized variant *SCHEDULED*. The results are sent back to the PC, analyzed and finally displayed.