DEMO: TYNDALL HETEROGENEOUS AUTOMATED WIRELESS SENSORS (THAWS)

TYNDALL NATIONAL INSTITUTE
University College Cork

Green sEnsor NEtworks for Structural monItoring
GENESI develops structural health monitoring systems for critical infrastructures such as tunnels, bridges, dams, private and public buildings, providing cutting edge green wireless sensor networks technology

KEYWORDS: structural health monitoring, energy harvesting, wireless sensor networks

First Workshop

11th March, 9.30-17.00
Hilton – Amsterdam Airport Schiphol

Introduction

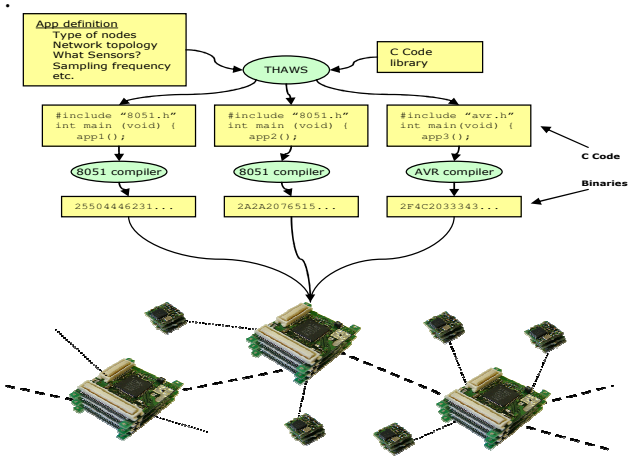
A wireless sensor network (WSN) may be constructed of heterogeneous nodes, or 'motes' – which may exhibit differences with respect to their physical composition (microcontroller, radio transceiver, etc.), or functionality (full function, leaf node, etc.) within a network, each of which collaborate as member nodes of the same application.

Programming of WSN devices traditionally requires each one to be programmed, or 'flashed', individually prior to deployment. During this process, the application (i.e. reading of sensors, sampling rates, etc.), identifiers, networking primitives, security settings, and so on, are all loaded into the memory of the device – having been compiled and built into a single executable file (binary or hex) that is then loaded.

Once the network is deployed, remote reconfiguration, or even redeployment, is known to be troublesome and expensive – particularly for large deployments. If, for example, the owner of an application wishes to reconfigure a minor parameter – such as the sampling rate, or parent ID – of a device that is a number of hops away from the basestation, or gateway node, it makes little sense to distribute a large binary file – potentially kB in size – for a minor update. As such, it is useful to create a simple mechanism whereby any type of update can be created and distributed, in a simple manner, to any node in the network, at minimum cost with respect to energy consumption and complexity. The Tyndall Heterogeneous Automated Wireless Sensors (THAWS) tool is designed to meet this requirement.

THAWS:
The THAWS tool makes use of a modular code library developed for Tyndall wireless sensor devices. It is implemented as a *python* script that automatically generates optimised code updates for in-network reprogramming of deployed nodes/motes.

.



Information Society
European Commission

This project is co-financed by the EC 7th Framework Programme on Research & Development, and part of the portfolio of the Embedded Systems Unit – G3 Directorate General Information Society
For more information please check:
http://cordis.europa.eu/ist/embedded

How is this done?

Python allows for the use of a number of existing modules – one of which is the JSON notation. This is a simple notation, with which the attributes of the entire network may be described. The Bsdiff algorithm enables the close analysis and comparison of two files (i.e. the old description of the network's attributes, and the new description), finding sections that partially match and highlighting sections that are different. Bspatch (another simple algorithm) is then used to create the new commands that are needed to reprogram the network – based on the output of the implementation of the Bsdiff algorithm. Delta encoding is used to compress the data to be transmitted – significantly reducing the overall size. It has been shown that using this encoding technique, impressive compression ratios can be achieved for small parameter alterations. For example, to change the sampling rate of a node a compression ratio of 0.48% can be achieved. In other words, what would have been 2896 bytes of code can be compressed to 14 bytes –achieving the same goal. The compression ratio for larger changes, such as a new application, is more modest; around 87%.

What will be demonstrated:

- The JSON notation used to describe the network (old and new)
- The THAWS tool generating updated exectutable files and compressed updates
- Generated binary files being flashed to the Tyndall motes
    - The Tyndall motes used in the demonstration will be of the following type:
        - 25x25mm Atmega 128/Nordic nRF905 configuration
        - 10x10mm Nordic nRF9E5 (includes 8051 MCU, 433 MHz TRX)
        - Respective USB programming boards

Tentative:

- *Code updates sent wirelessly to the mote.*

Limitations:

Propagation throughout a network remains to be unresolved. Integration with algorithms/solutions such as Trickle or Deluge would be useful. The process, as will be demonstrated, remains to be largely manual in nature: whilst the JSON notation is simple, a more user friendly tool can be developed.

THAWS for GENESI:

What is needed?

- New code library needs to be developed for the GENESI nodes (new microcontroller, radio transceiver components)
- Enhanced automation – it remains to be a relatively manual procedure
- Support for in-network commands

What is desirable?

- GUI?
- OS support – Contiki, TinyOS?
- Investigation and integration with existing mechanisms used in wireless sensor networking- such as Deluge, Drip, Trickle, etc.