Some thoughts and feedback from extensive experimentations with a large scale IoT testbed

Franck Rousseau
Université Grenoble Alpes
Grenoble INP – LIG Lab
Franck.Rousseau@imag.fr
Credits

• Isabel Vergara, PhD now at CEA LETI

• Ana Bildea, PhD now at Arago Systems

• Michał Król, PhD student

• Andres Cantor, João Zeni, Antoine Ubrich, Yongkan Huang, and other interns

• Étienne Dublé, Olivier Alphand, Andrzej Duda

• Quentin Lampin, Frédéric Evennou, Dominique Barthel, Orange Labs

• Many others
Our usage these last 4 years

- Very early users
  - Things have changed a lot since then
- Projects funding our work
  - ANR Aresa2, ANR Iris, ICT Calipso
  - Wireless sensor networks related obviously
- Software we have used
  - Contiki OS, TinyOS, FreeRTOS, bare bone
Keep in mind

• We focus on large scale experiments
  • Several hundred nodes
  • Running for days
  • Generating loads of data
• Senslab / IoT-lab is very good at that
  • But even with such a great tool, we found that it remains a challenge to do it well
Experiments we made

• Routing
  • Waypoint routing, Featurecast, (comparison with previous work)

• MAC protocols
  • Wake on Idle, Sleep on Idle, (comparison with previous work)

• Transmissions
  • Bit Error Rate and Packet Error Rate analysis
Results?

- It worked, we made publications, deliverables, etc.
- But since we learn from our mistakes, let's focus on where we struggled, what was missing, and what has failed.
Routing

- Study mechanisms to route packets in the networks
- Long range transmissions relative to testbed range
- Fully connected network, single hop, no routing!
Large topologies

- We expect many hops to test routing protocols
- Depends on the transmission range
- TX power reduction, impact on experiments?
Traffic

- Generate traffic patterns, distributions
  - No advanced arithmetics in MCUs
  - Coordination among nodes
  - Create defects

Remote control nodes
“send frame”
MAC protocols

- Analog signaling for energy efficient synchronized communications
  - No need to decode frames, only sensing
  - Need very flexible radios like CC1101 to experiment some ideas
Wake on Idle

- Neighbor maintenance and medium access
- Pair of nodes tracks each other using analog signaling at pseudo-random instants
- Medium access using code violation

![Diagram showing Node A, B, C, and D, with time on the x-axis and different symbols indicating code violation, frame coming next, and pseudo-random instants.]

11
Complex implementation

• Remote debug is impractical in the dev. phase

• Local debug on desktop is very limited and very bulky
  • 2-3 nodes max and as many serial interfaces for debug control and traces

• Port / adapt previous work for comparison purposes
  • Quite tedious
Channel characterization

• Study Bit Error Rate
  • Use CC1101 infinite mode to generate very long sequences

• Study Packet Error Rate
  • Relation between BER and PER
  • Derive packet receive ratio from RSSI and LQI
BER experiments

• Very large scale
  • Send from each node to all the others
    • x100 receivers
  • Serial link way too slow compared to radios
    • Use reduced wireless bitrate
  • 450 GB of traces collected — raw decoded bits
Idea

- Transmit pseudo-random sequence cycle
- Analyse received data to detect bit errors
- Collect other data periodically for eventual further analysis
  - RSSI, temperature, …

TX

- PR sequence
- cycle PR sequence
- embed and protect sequence number

RX

- bit sequence
- locate PR sequences
- extract sequence number
- compute BER
Raiders of the lost bits

Node A TX
011101101001110110

Node B RX
011101101011101101

Node C RX
011101101001110110

Node D RX
011101101101110110

BER ≈ 10^{-7}

BER is huge
One or several bits missing

BER = 0
Automation

- Large scale means
  - A lot of runs, with varying parameters
    - Scriptable experiments
  - Traffic generation, synthetic behavior, perfect behavior
    - Remotely controlled generic firmware
  - Random node crashes
    - Resume experiments
- Experiment control with a Python framework
Experiment setup

• Debugging low level functions on the desktop is fine
  • Usual stuff: JTAG, SWD, debugger, etc.

• Debugging and validating an entire protocol / application on a remote platform is tedious
  • No direct access, node reservation
  • No easy integrated build and test

• Need for a small size platform: WalT
Traces and visualization

- Collect
  - Events in nodes
    - Instrumented Contiki OS
    - All the traffic in the network
    - Need fast links from nodes to the infrastructure
  - Visualize traces with usual tools
    - VizWalT plugin for Cooja
    - Inject real traces in Cooja
- Full trace collected from nodes
Thanks