Testumgebungen für das verteilte Internet der Dinge

Institut für verlässliche Embedded Systems und Kommunikationselektronik (ivESK)

Dipl.-Inform. (FH) Manuel Schappacher
Prof. Dr.-Ing. Axel Sikora Dipl.-Ing. Dipl. Wirt.-Ing.
Institute of Reliable Embedded Systems & Communication Electronics (ivESK)

• Research & Pre-Development Projects
  o Networking/Communication
  o Hardware close firmware development
  o Stack development
  o Tool development

• Industrial Projects
  o Analytical studies
  o System Design & Implementations
Ch.2: Test Methods

• Several steps to develop and test (wireless) communication systems depending on different aspects
  o New protocols or technologies
  o Existing technology with new features
  o Test of existing protocols

• How does a network behave
  o Depending on the network load
  o With many nodes
  o With interferers
Ch.2.1: Test Methods - Simulation

- **Simulation** often is used as an initial step to develop wireless networks
  - Especially for new technologies or features
  - Prove of concept and evaluation

- Allows high scalability
  - Several nodes within a single simulator
  - Simulation of long-time scenarios within a short period of real time

- Very good analysis methods
  - Statistics
  - Debugging possibilities
  - Interfaces to 3rd party tools
Ch.2.1: Test Methods - Simulation

- Based on calculations
  - e.g. using MATLAB models
- Network simulator based
  - Riverbed Modeler\(^1\) (former OPNET Modeler)
  - ns-2\(^2\) and ns-3\(^3\)
  - OMNet++\(^4\)

- Creation of a separate code-branch (often) required
  - Due to different paradigms
  - Due to insufficient abstraction layers

\(^1\)http://www.riverbed.com/de/products/steelcentral/opnet.html?redirect=opnet
\(^2\)http://www.isi.edu/nsnam/ns/
\(^3\)https://www.nsnam.org/
\(^4\)https://omnetpp.org/
Ch.2.2: Test Methods - Virtualization

- **Virtualization** helps testing original code branches of devices on a single PC as virtual nodes
  - (Nearly) unlimited resources with regard to the number of devices and with regard to topologies
  - Easy debugging and analyzing (compared to embedded development environments)

- Test and verification of original (embedded) code

- **Virtual Network**
  - Access to network behavior
  - Virtual topology
  - Channel characteristics
Ch.2.2: Test Methods - Virtualization

- **VTENN** (Virtual Test Environment for Networking Nodes)
  - Host provides
    - Container for virtual nodes (N)
    - Virtual network (VNET)
    - Connection to the host network (HNET)
  - Possibility to virtualize a complete system
    - Including test environment
    - Monitoring
    - Backend applications

LINUX HOST

- TOOLS
  - IDE
  - TTCN
  - TEST
  - CTRL
  - MONITOR

- HNET
  - BA
  - ER (G)

- VNET
  - VR N
  - VR N
  - VR N

- virtual node
- published channel
- subscribed channel

- virtual node
- published channel
- subscribed channel
Ch.2.2: Test Methods - Virtualization

• Requires HAL adaptation to the virtualization host
  o Same as porting to another embedded platform
  o Virtual communication driver

• Requires virtual network
  o Inter process communication (IPC) used for network establishment
  o Controllable topologies

• Requires virtual channel model
  o packet error rate (PER)
  o attenuation
  o delay
Ch.2.2: Test Methods - Virtualization

- **Network Manager**
  - Receives Commands from a Test Executor (TE) e.g. to connect 2 nodes via 1 channel

- **Node**
  - Provides interface to the Node Under Test
  - Transmits and receives data to/from a channel

- **Channel**
  - Forwards packets between 2 nodes
  - Performs packet manipulation
    - Channel characteristics
    - Data Content

- **Ports** allow to control the elements and provide communication among nodes
Ch.2.3: Test Methods - Emulation

- **Test original embedded firmware on real target devices using real radio**
  - Test behavior with regard to hardware based effects (e.g. timing issues)
  - Execution of integration and regression test

- **Emulation** of the radio channel
  - Without emulation the behavior of a radio channel is unpredictable and can hardly be reproduced
Ch.2.3: Test Methods - Emulation

• Automated testing environment with real networking nodes called “Automated Physical Testbed (APTB)"

• Topologies and RF environment are controllable and predictable
  - Wired connection of RF elements
  - Antenna in-/outputs to RF waveguides

• Isolated, well controlled, repeatable tests in various conditions
  - Statistical analysis
  - Regression tests
Ch.2.3: Test Methods – Emulation (APTB Components)
Ch.2.3: Test Methods – Emulation (APTB Versions)

V1
Physical testbed for verification of routing mechanisms (EnOcean)
Static Topology
Static Devices

V2
Physical testbed for the Ko-TAG system (Automotive, VRU protection)
Dynamic Topology (4 Nodes)
Controllable Devices
SCPI Interface

V3
Physical testbed for Wireless Devices
Dynamic Topology (12 Nodes)
Controllable Devices
MATLAB Interface (SCPI Based)
Web Interface

V4*
Physical testbed for IoT
Dynamic Topology (12 Nodes)
Controllable Devices
TTCN Interface
VTENN Compatible
Ch.2.3: Test Methods – Emulation (APTB Versions)
Ch.2.3: Test Methods – Emulation (APTB Architecture)
• Control software for RF elements
  o SCPI (Standard Commands for Programmable Instruments)
  o Low-Layer Commands to directly access RF elements of the APTB e.g:
    APTB:RADElements:ATTenuator:SETAttenuation <id>,<val>
    APTB:RADElements:SWitch:SETEnable <id>,<val>
    APTB:NETEElements:NOde:SENDData <id>,<data>
  o Interpreted an executed by the APTB Controller which includes an according SCPI parser/handler

• MATLAB scripts to automate the test cases
  o Scenarios can be described in human readable manner (CSV, XML)
  o Uses Java-IO libraries to send the SCPI commands to the controlling devices.

• Ongoing TTCN integration
Testing and Test Control Notation (TTCN-3)

- TTCN-3 is a scripting language with strong typing used in protocol compliance and interoperability testing
- TTCN-3 is used for 3GPP, LTE and automotive protocols
- Abstract definition of test-cases, independent from SUT

```c
// Testcase TC_resolveEtsiWww() runs on DnsClient
{
    timer t_ack;
    srvPort.send(m_dnsQuestion("www.etsi.org"));
    t_ack.start(1.0);
    alt {
        []srvPort.receive(mw_dnsAnswer("172.26.1.17")) {
            setverdict(pass);
        }
        [] srvPort.receive { // any other message
            setverdict(fail);
        }
        [] t_ack.timeout {
            setverdict(inconc);
        }
    }
    t_ack.stop;
}
```
Ch. 4: summary & outlook

• Development and verification of communication systems require a chain of different tools

• All the different steps are already used within ivESK
  o Using existing tools as well as own developments

• Currently ongoing fusion of the different tools
  o All controlled by TTCN test framework
  o Usage of common test cases e.g. for virtualization and emulation
Q&A