

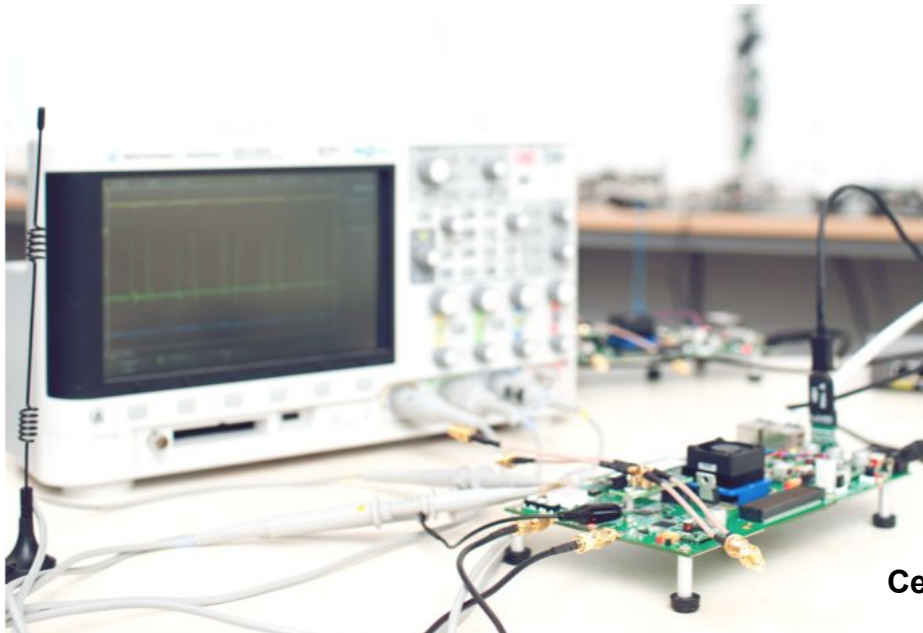
Secure Mobile Networking Lab Exercise / Project



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Summer 2016

Kick-off meeting - Topic Overview



Prof. Dr.-Ing. Matthias Hollick
matthias.hollick@seemoo.tu-darmstadt.de



Prof. Dr.-Ing. Matthias Hollick

Technische Universität Darmstadt
Secure Mobile Networking Lab - SEEMOO
Department of Computer Science
Center for Advanced Security Research Darmstadt - CASED

Mornewegstr. 32
D-64293 Darmstadt, Germany
Tel.+49 6151 16-70922, Fax. +49 6151 16-70921
<http://seemoo.de> or <http://www.seemoo.tu-darmstadt.de>

Topic Overview

MW1: Replacement of BGP Routers (2 - 3 Students)

MW2: STEAN OS (2 - 4 Students)

MW3: Greetings from the Network (1 Student)

MW4: Virtualizing the Testbed (2 - 4 Students)

JC1: Android Signal Analyzer (1 - 4 Students)

JC2: Visible Light Communication (2 - 4 Students)

DS*2: Simulating Highly Directional 60 GHz Mesh Networks
(2 - 4 Students)

MiS*1: WMN Inspect: Implement a network traffic and
topology analyzer (2 - 4 Students)

MiS2: Implementing 'ping' and 'traceroute' for Castor
(1 - 2 Students)

MiS3: Replacing Serval's Mesh Routing with a Secure
Protocol (2 - 3 Students)

MiS4: Click2iOS (1 - 2 Students)

MaS1: Implementing 802.11 on USRPs (2 - 4 Students)

MaS2: Develop the SEEMOO software-defined radio
(4 - 8 Students)

MaS3: WiFi Firmware Hacking (2 - 4 Students)

DY1: Implementing and Improving Constructive
Interference with SDR (1 - 2 Students)

DY2: Data Collection with TSCH (2 - 3 Students)

AB1: Implementing aDTN on Click (2 - 4 Students)

FA1: Framework for supporting multiple wireless
technologies on Android (2 Students)

FA2: Analysing energy consumption on android
(1 - 2 Students)

RK*1: Time-Synchronized WiFi for WARP (2 - 4 Students)

RK2: Decoding Wireless Collisions (2 - 3 Students)

RK3: GPS-based 3D Compass (2 - 4 Students)

RK*4: ns-3 Channel Model for WMNs (2 - 4 Students)

LA*1: High-Level Testbed Framework (2 - 4 Students)

MF1: What is up with your WhatsApp? (2 Students)

AA1: Prototyping 5G using LabView SDR Platform
(3 - 4 Students)

Your tasks:

- 
- TECHNISCHE
UNIVERSITÄT
DARMSTADT

STEAN OS (MW2)

STEAN is a system to manage the context of a large number of networked systems such as router, network functions or protocols. It currently runs on any Unix like operating system.

We now would like to port STEAN to MiniOS, a minimalistic operating system running on top of the XEN hypervisor. This will help us to quickly scale the number of instances depending on the current load.



Your tasks:

- Get familiar with MiniOS and STEAN
- Migrate the STEAN code to plain C (or write appropriate helpers)
- Port STEAN to MiniOS

Type: Lab/Project for 2-4 students

Prerequisites:

C/C++, XEN Hypervisor

Contact:

Marc Werner

Greetings from the Network (MW3)

Wireless Mesh Networks use periodic messages (HELLOs) to detect other nodes. These messages are sent at a fixed interval even if the topology does not change. The question answered with this lab is: *How can we reduce the amount of HELLO messages?*



Your tasks:

- Get familiar with the Click Modular Router Framework
- Analyze the existing routing protocols within Click
- Implement a dynamic HELLO strategy to reduce network load

Type: Lab for 1 student

Prerequisites:

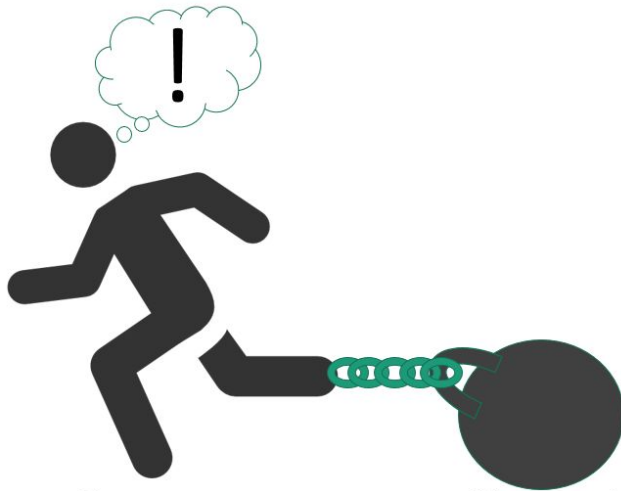
C/C++, Mesh Networking

Contact:

Marc Werner

Virtualizing the Testbed (MW4)

We now live in a virtualized world but our Wireless Testbed still runs all software on bare metal. Your task is now to virtualize the nodes in our testbed but still allow access to the physical hardware (wireless card, SDR etc.) where necessary.



Your tasks:

- Gather the requirements of typical testbed users
- Select a suitable virtualization platform
- Setup a prototypical node with the selected software
- Create virtual machines based on the requirements above, that show the feasibility of your approach

Type: Lab/Project for 2-4 students

Prerequisites:

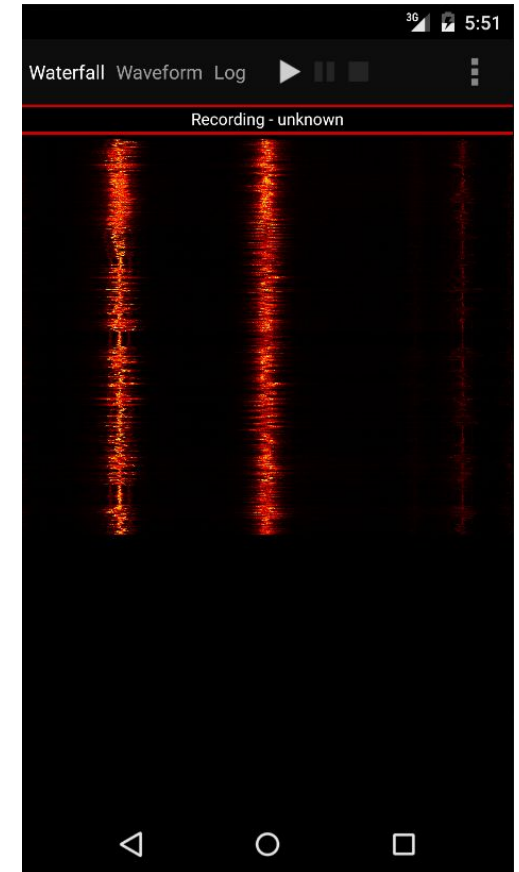
Knowledge of Virtual Machine environments

Contact:

Marc Werner

Android Signal Analyzer (JC1)

Walking through the city and inconspicuously sniffing wireless protocols - this can be established with our Android signal analyzing software, written in Java. However, some features are still missing...



Your tasks:

- Decode bits from standard modulation schemes like BPSK.
- Enhance support for the rad1o SDR.
- Choose additional features you like depending on the group size.

Type: Lab/Project for 1-4 students

Prerequisites:

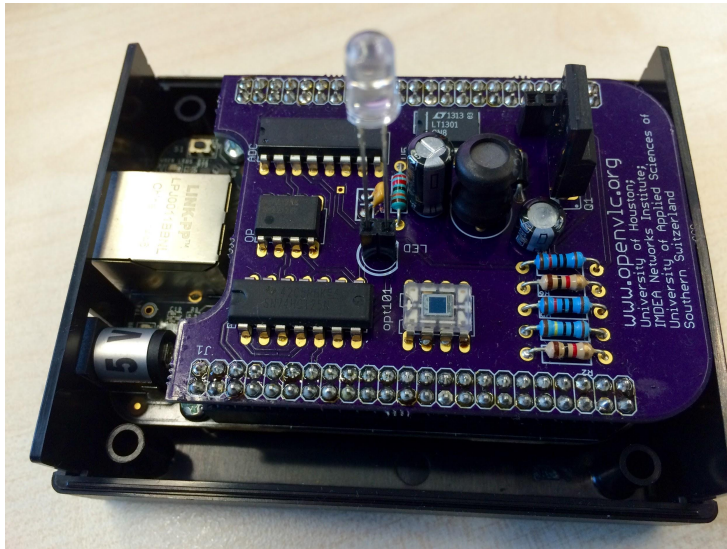
Android / Java, Signal Processing

Contact:

Jiska Classen

Visible Light Communication (JC2)

We have two OpenVLC prototype boards. Depending on the team size, there are multiple tasks.



Your tasks:

- Implement additional modulation schemes in the existing C driver.
- Implement a modulation via MATLAB (connections to the BeagleBone Black are already supported in MATLAB).
- Make a performance analysis on the hardware limits.

Type: Lab/Project for 2-4 students

Prerequisites:

Good C or MATLAB programming skills

Contact:

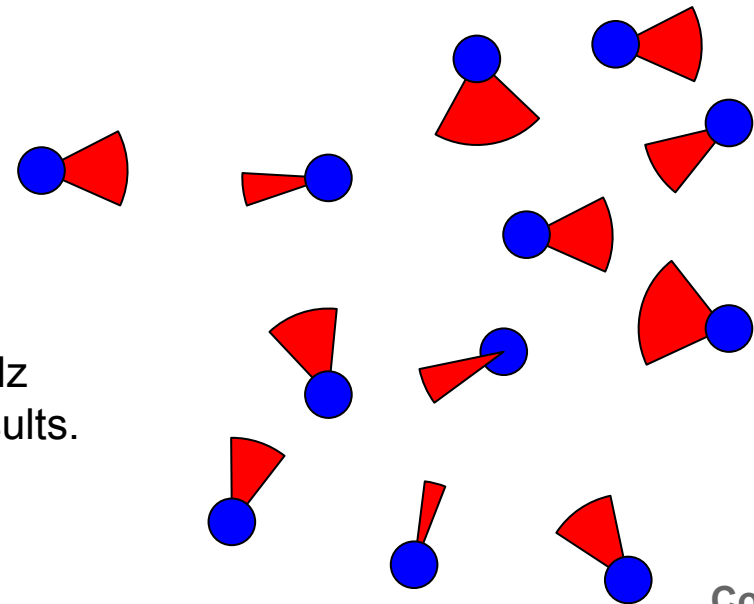
Jiska Classen

Simulating Highly Directional 60 GHz Mesh Networks (DS*2)

Using millimeter-waves in mesh networks promises advantageous features as increased throughput and low interference. Directional transmission allow for simultaneous communications since signals are directly steered towards their destination. However, the challenge lies in finding the optimal antenna orientation for neighbour discovery. We aim to investigate this problem in ns-3 simulations.

Your tasks:

- Extend our ns-3 implementation
- Investigate different neighbour discovery strategies
- Evaluate the performance compared to omni-directional networks
- Outline advantages and challenges of 60 GHz mesh networks based on your simulation results.



Type: Lab/Project for 2-4 students

Prerequisites:

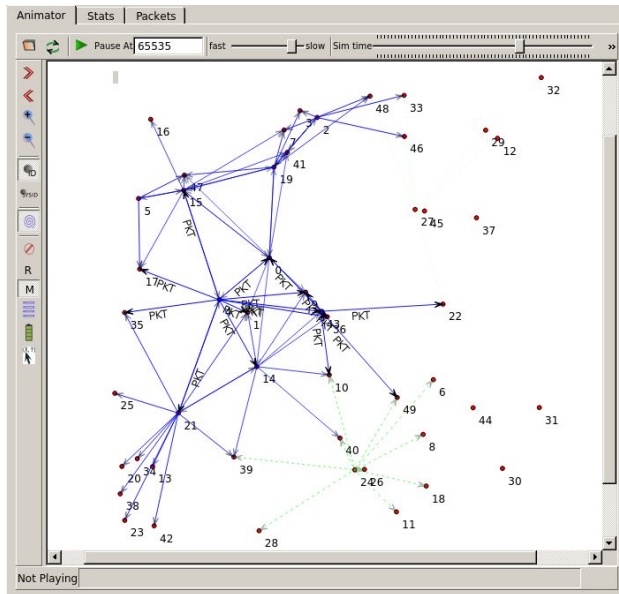
C/C++, Wireless network basics

Contact:

Daniel Steinmetzer, Milan Schmittner

WMN Inspect: Implement a network traffic and topology analyzer (MiS*1)

Debugging distributed system is a hard task. It is even harder to do this in dynamic and error prone wireless systems.



Your tasks:

- Build a tool that can display a network topology with given node positions similar to NetAnim (left)
- The traffic input is given as Wireshark-compatible PCAP files for each individual node
- Support for Wireshark/Tshark filters
- Sync “Big Picture” view with (multiple) Wireshark views, such that it is easy to zoom in and investigate at packet level

Type: Lab/Project for 2-4 students

Prerequisites:

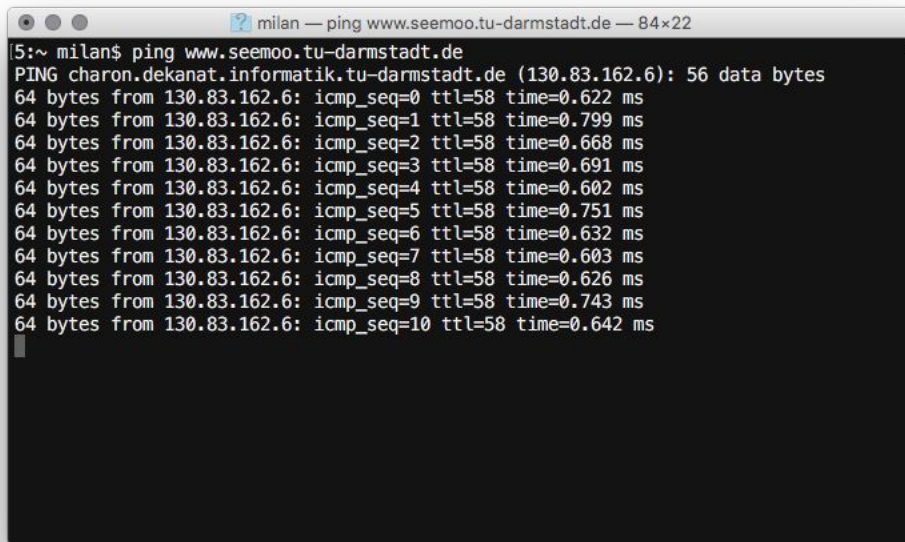
Wireshark, “suitable” programming language

Contact:

Milan Schmittner, Marc Werner

Implementing 'ping' and 'traceroute' for Castor (MiS2)

Castor is a secure routing protocol for mobile ad hoc networks. The current implementation (C++) runs as a layer 2 protocol, such that external network debugging tools, for example, 'ping' and 'traceroute', will only see a single hop.



```
milan — ping www.seemoo.tu-darmstadt.de — 84x22
5:~ milan$ ping www.seemoo.tu-darmstadt.de
PING charon.dekanat.informatik.tu-darmstadt.de (130.83.162.6): 56 data bytes
64 bytes from 130.83.162.6: icmp_seq=0 ttl=58 time=0.622 ms
64 bytes from 130.83.162.6: icmp_seq=1 ttl=58 time=0.799 ms
64 bytes from 130.83.162.6: icmp_seq=2 ttl=58 time=0.668 ms
64 bytes from 130.83.162.6: icmp_seq=3 ttl=58 time=0.691 ms
64 bytes from 130.83.162.6: icmp_seq=4 ttl=58 time=0.602 ms
64 bytes from 130.83.162.6: icmp_seq=5 ttl=58 time=0.751 ms
64 bytes from 130.83.162.6: icmp_seq=6 ttl=58 time=0.632 ms
64 bytes from 130.83.162.6: icmp_seq=7 ttl=58 time=0.603 ms
64 bytes from 130.83.162.6: icmp_seq=8 ttl=58 time=0.626 ms
64 bytes from 130.83.162.6: icmp_seq=9 ttl=58 time=0.743 ms
64 bytes from 130.83.162.6: icmp_seq=10 ttl=58 time=0.642 ms
```

Your tasks:

- Implement 'ping' for Castor
- Implement 'traceroute' for Castor
- Leverage on Castor's inherent ACK mechanism

Type: Lab/Project for 1-2 students

Prerequisites:

C/C++, network programming

Contact:

Milan Schmittner

Replacing Serval's Mesh Routing with a Secure Protocol (MiS3)

Serval allows communication between mobile devices without any infrastructure. It has a dual-stack architecture: (1) a classic mesh routing protocol for 'real-time' applications such as speech; and (2) a bundle protocol for delay tolerant message delivery. Castor is a secure routing protocol for mobile ad hoc networks and could be used as a drop-in replacement for 'real-time' routing.

Your tasks:

- Familiarize yourself with Serval's mesh routing codebase
- Identify interfaces to other Serval components
- Transparently substitute Serval's mesh protocol with Castor (other functionality in Serval should not be impeded)



Type: Lab/Project for 2-3 students

Prerequisites:

C (!)/C++, network programming

Contact:

Milan Schmittner

Click2iOS (MiS4)

iOS exposes the so called Multipeer Connectivity Framework (MCF) which enables direct device-to-device communication. Click is a modular routing framework written in C++ and is currently used to implement routing protocols for wireless multihop networks. To run the same Click code on iOS, we need to interface with MCF.

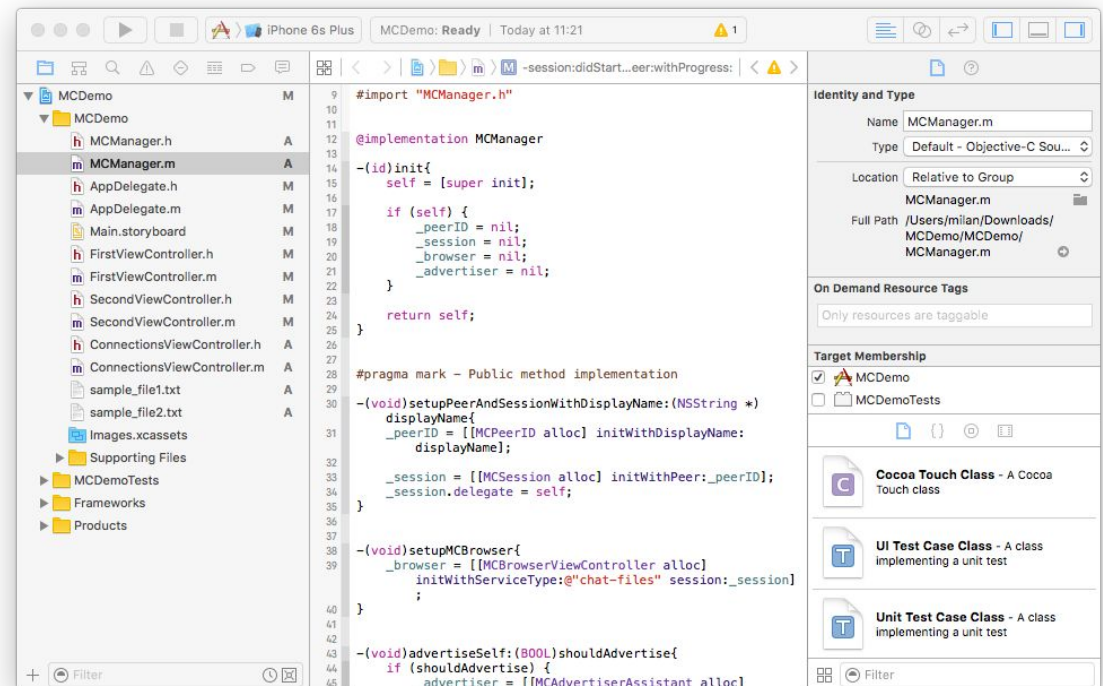
Your tasks:

- Familiarize yourself with Click
- Implement interfacing elements with MCF in Click (*PeerID, Neighbors, IO elements, ...*)
- Write an iOS library which exports your Click integration to other Apps
- Implement a demo App using your new library

Type: Lab for 1-2 students

Prerequisites:

Objective-C, C++



Contact: Milan Schmittner

Implementing 802.11 on USRPs (MaS1)

There is already an 802.11 reference design for WARPs. We intent to have something similar for USRPs, which is a different software-defined radio platform.



Your tasks:

- Implement the 802.11 PHY for the USRP.
- Implement the 802.11 MAC for the USRP.
- Show that your implementation can communicate with standard 802.11 hardware.



Type: Lab/Project for 2-4 students

Prerequisites:

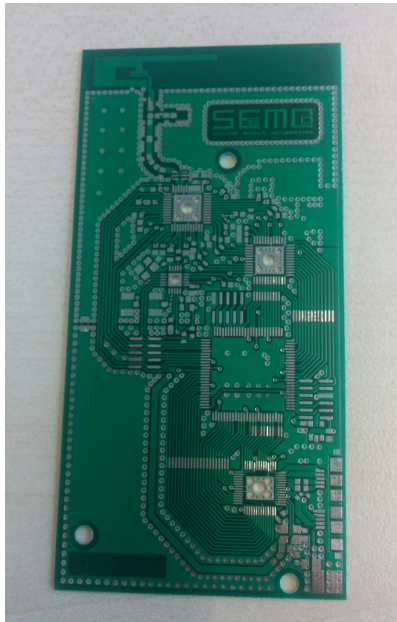
C, Embedded Systems, FPGA, Signal Processing

Contact:

Matthias Schulz

Develop the SEEMOO software-defined radio (MaS2)

We already designed the hardware platform for a cheap (approx. 30 EUR) software-defined radio (SDR) that can transmit and receive in the 2.4 GHz band. You are support to develop the software to interface the SEEMOO SDR.



Your tasks:

- Get the USB 3 FIFO interface up and running.
- Program the FPGA to exchange samples between the ADC/DAC and the USB 3 FIFO.
- Develop a driver to configure the hardware components to change frequencies, bandwidths, ...
- Depending on the team size, partial solutions are fine.

Type: Lab/Project for 4-8 students

Prerequisites:

C, Embedded Systems, FPGA, Electrical Engineering

Contact:

Matthias Schulz

WiFi Firmware Hacking (MaS3)

After reverse engineering the BCM4339 chipset in the NEXMON project [1], we now want to implement applications directly on the WiFi chip and extend its functionality. You chose one of the following or propose your own topic:



Possible Topics:

- Implement an advanced Ad-Hoc mode
 - Implement an Ad-Hoc implementation that allows to change modulation schemes on a per frame basis
- Implementation of WiFi chip malware
 - Implement a proof of concept malware that can reside in the WiFi chips firmware
- Manipulation of real time parameters
 - Implement a firmware modification that can influence timings and that can transmit at arbitrary points in time

Type: Lab/Project for 2-4 students

Prerequisites:

C, ARM Thumb Assembler, Embedded Systems, IDA Pro

[1] <http://nexmon.org>

Contact:

Matthias Schulz

WiFi Firmware Hacking 2 (MaS3)

After reverse engineering the BCM4339 chipset in the NEXMON project [1], we now want to implement applications directly on the WiFi chip and extend its functionality. You chose one of the following or propose your own topic:



Possible Topics:

- Implement a GDB Debug Interface for the ARM core
 - We already found the DBG Registers, you need to interface them to GDB
- Porting the Firmware Hack to Raspberry Pi 3
 - The RasPi3 contains a BCM4343 WiFi chipset, which might benefit from NEXMON, too ;-)

[1] <http://nexmon.org>

Type: Lab/Project for 2-4 students

Prerequisites:

C, ARM Thumb Assembler, Embedded Systems, IDA Pro

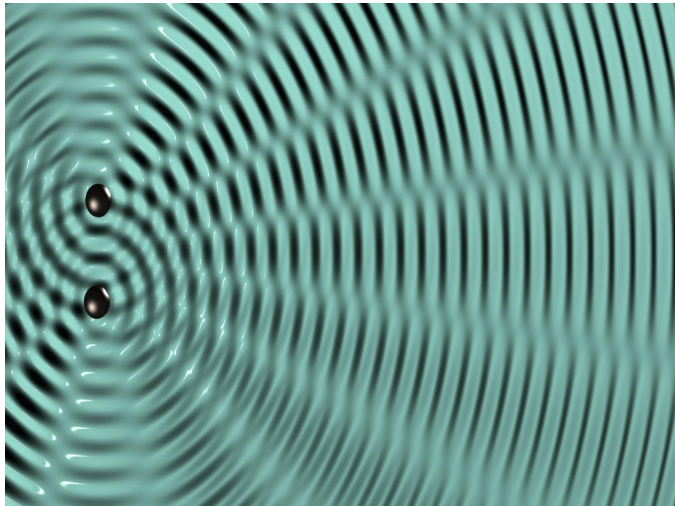
Contact:

Matthias Schulz

Implementing and Improving Constructive Interference with SDR (DY1)

Constructive Interference (CI) is used to significantly improve QoS of WSN communication.

However, It is hard to achieve on standard radio chips.



Your tasks:

- Implement 802.15.4 communication on USRP
- Implement CI on USRP
- Improve CI by realizing better/longer DSSS code



Type: Lab/Project for 1-2 students

Prerequisites:

C programming, Signal Processing

Contact:
Dingwen Yuan

Data Collection with TSCH (DY2)

Data Collection is the most important application of WSN. However, it is non-trivial to make it low-latency, high-reliability and energy-efficient. The newly appeared 802.15.4e (TSCH, Time Slotted Channel Hopping) standard provides a potential solution.

Your tasks:

- Implement on OpenWSN OS, which provides TSCH implementation.
- Use TSCH features: multi-channel, TDMA.
- Topology control: form tree structure, support node join and leave
- Auto scheduling: a frame is composed of two parts -- reserved and contention subframes. Reserved subframes give each source one chance to send to sink. Contention subframes compensate for any losses.



Type: Lab/Project for 2-3 students

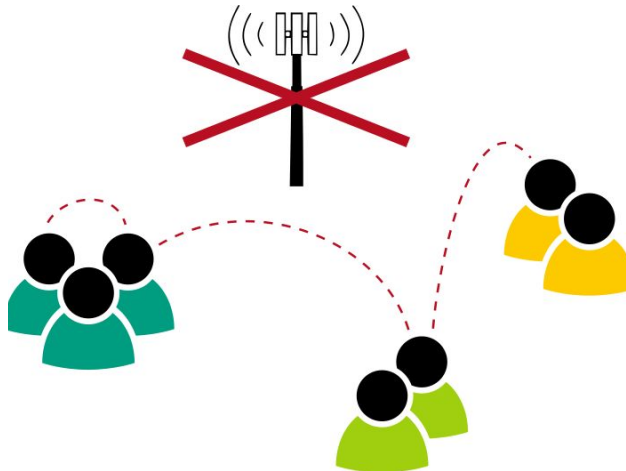
Prerequisites:

C programming

Contact:
Dingwen Yuan

Implementing aDTN on Click (AB1)

aDTN is a secure network layer protocol for wireless networks that features strong privacy protection, such as sender and recipient anonymity and unobservability of communications.



Your main task is to implement 4 variants of the aDTN protocol:

- 2 different protocol versions
- using 2 different sets of cryptographic protocols

The implementations are to be done on the Click modular router (in C++ and Click script), and finally tested with the NS3 network simulator.

Type: Lab/Project for 2 - 4 students

Prerequisites:

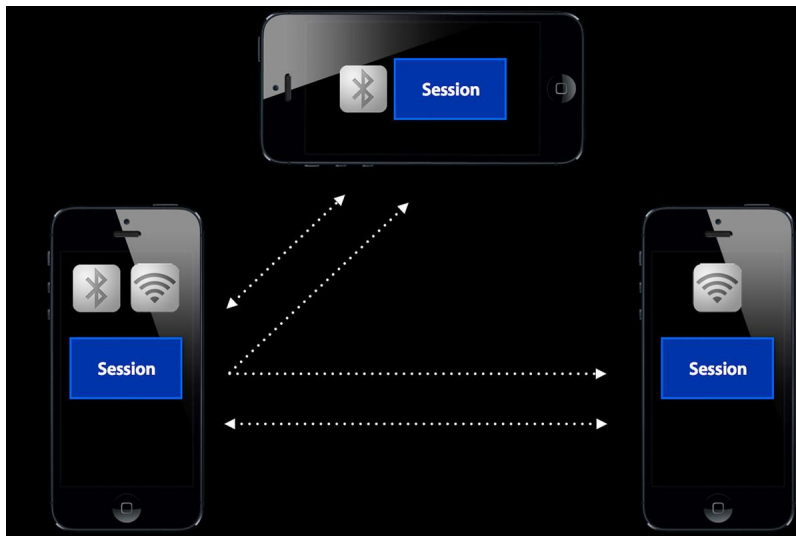
Knowledge of C++ (preferred) or C

Contact:

Ana Barroso

Framework for supporting multiple wireless technologies on Android (FA1)

Multipeer Connectivity Framework in iOS provides an abstraction layer to allow a communication between nearby devices by using multiple wireless technologies such as Bluetooth, peer-to-peer Wi-Fi or Wi-Fi infrastructure.



Source: <http://www.imore.com/multipeer-meshed-networks-and-why-risk-managers-will-love-them>

Your tasks:

- Implement an abstraction layer in Android to allow nearby devices to establish a connection between devices using an available interface
- The framework should automatically choose an interface communication (Wi-Fi direct, or Bluetooth) according to the available interface
- The process of discovering and connection between devices should be abstract to the application layer
- Create an example App using your implemented framework

Type: Lab/Project for 2 students

Prerequisites:

Android/Java, Wireless Communications

Contact:

Flor Álvarez: falvarez@seemoo.tu-darmstadt.de

Analysing energy consumption on android (FA2)

Saving energy on smartphone devices is considered one of the most important issues of those devices. Due to rapidly grow of energy consumption sources in smartphones, it is important to analyze and monitor their energy consumption.



Source: <http://media02.hongkiat.com/conserv-smartphone-battery-life/conserv-battery-life.jpg>

Your tasks:

- Analyze the existing energy consumption sources on smartphone devices. How much energy is necessary e.g. in devices associations, key generation
- Provide an android application in order to evaluate the power consumption of at least 2 different:
 - encryption techniques
 - sensor data reading e.g. GPS
 - wireless communication technologies e.g WiFi-Direct

Type: Lab/Project for 1-2 students

Prerequisites:

Android/Java, Wireless Communications

Contact:

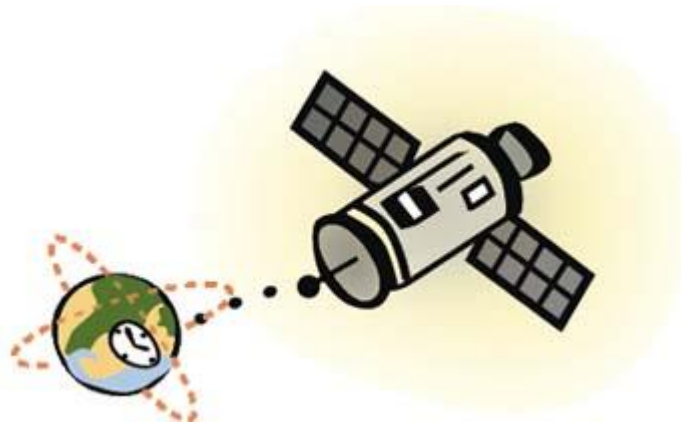
Flor Álvarez: falvarez@seemoo.tu-darmstadt.de

Time-Synchronized WiFi for WARP (RK*1)

Medium access control schemes:

- Most widespread in current networks (e.g. IEEE 802.11): CSMA/CA
- In some scenarios very promising: slotted and reservation based schemes

In this project: Devise and implement a **slotted medium access control** scheme that is based on **GPS synchronization!**



Your tasks:

- Select a suitable GPS module
- Develop an interface to connect the GPS module to the WARP (e.g., microcontroller)
- First: Trigger transmissions/receptions in WARPLab
- Later: Integrate your slot-based medium access control scheme into WARP's 802.11g MAC
- Evaluate the performance of your implementation

Type: Lab/Project for 2-4 students

Prerequisites:

C/C++, Hardware Design (FPGA), Embedded Systems

Contact:

Robin Klose, Matthias Schulz

Decoding Wireless Collisions (RK2)

Collisions on wireless networks:

- typically lead to packet loss
- but may still reveal useful information

In this project: Extract information from collisions!



Your tasks:

- Review related work
- Create an experiment with WARP SDRs
- Devise and evaluate methods to reconstruct collided IEEE 802.11 frames:
 - Detect frame alignments
 - Detect known data fields
 - Estimate content of unknown data fields

Type: Lab/Project for 2-3 students

Prerequisites:

Signal Processing, MATLAB, C/C++

Contact:

Robin Klose [robin.klose@seemoo.tu-darmstadt.de]

GPS-based 3D Compass (RK3)

In this project, you will devise a high-precision GPS-based 3D compass:

- Estimate the angle of arrival of GPS signals by means of antenna arrays
- Determine the satellite positions on the firmament
- Derive the device's orientation by combining both pieces of information



Your tasks:

- Review existing GPS compass solutions
- Devise theoretical models
- Design and implement a prototype with an SDR
- Evaluate the precision of your solution

Type: Project for 2-4 students

Prerequisites:

Signal Processing, MATLAB, Python, C/C++

Contact:

Robin Klose [robin.klose@seemoo.tu-darmstadt.de]

ns-3 Channel Model for WMNs (RK*4)

Many users of the network simulator **ns-3** focus on wireless simulations based on standards like IEEE 802.11, WiMAX, LTE, etc.

We want to devise a **new channel model** that leads to more **realistic** simulation results in **multi-hop** scenarios.



Your tasks:

- Get familiar with ns-3 programming
- Get familiar with established channel models for wireless channels
- Implement a new channel model that takes the peculiarities of wireless multi-hop networks into account (e.g., interferences by other transmitters)
- Evaluate your channel models and compare them to other built-in models of ns-3

Type: Lab/Project for 2-4 students

Prerequisites:

Signal Processing, C/C++, Python, MATLAB

Contact:

Robin Klose, Milan Schmittner, *

High-Level Testbed Framework (LA*1)

We are building a wireless mesh testbed for the NICER Emergency-Response Lab. It will span multiple sites (starting with 2 buildings) and use heterogenous hardware (from SDR to RPi3). To ease running experiments a testbed framework like OMF [1] should be integrated.



Your tasks:

- Get familiar with OMF or a similar framework
- Setup a functional prototype VM
- Integrate support for our low-level framework
- Implement new features
- Evaluate your work in our real-world testbed

Type: Lab for 2-4 students

Prerequisites:

Scripting (Ruby, Python), Linux

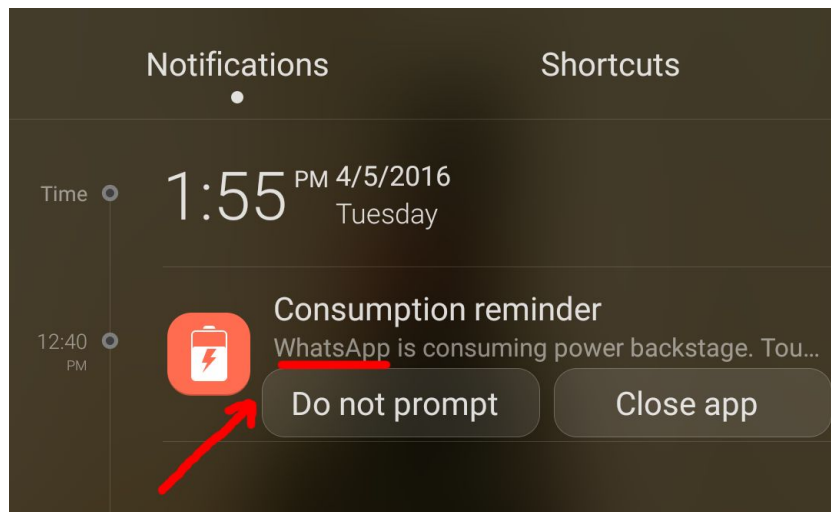
[1] <https://mytestbed.net/>

Contact:

Lars Almon, Matthias Krügl

What is up with your WhatsApp? (MF1)

WhatsApp is a messaging service that has attracted an enormous amount of users worldwide. It provides a rich set of capabilities and everybody just loves it. But what exactly does it do backstage? Does it spy on us? Some weird notifications appear. We are set out to discover it.



Your tasks:

- Get familiar with the WhatsApp protocol (previous reverse engineering endeavours)
- Utilize an ADB workaround on Android to escalate privileges and learn as much as possible about the WhatsApp
- Root an Android phone and perform the monitoring again in order to find out more about the WhatsApp components
- Integrate everything into an app for stealthy monitoring

Type: Lab/Project for 2 students

Prerequisites:

Android internals, Java, C/C++

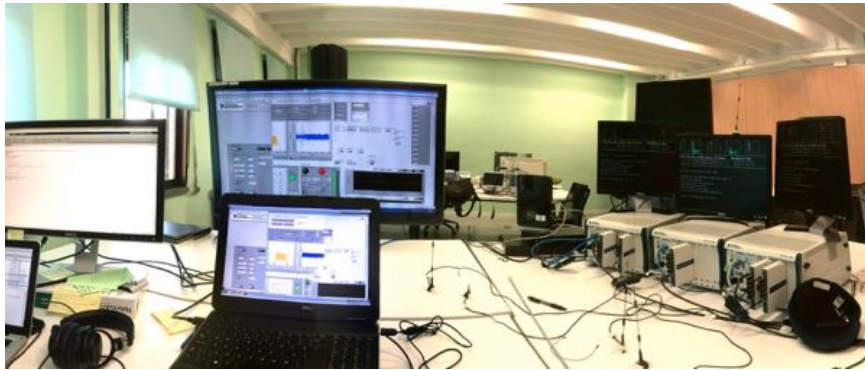


Contact:

Mikhail Fomichev [mfomichev@seemoo.tu-darmstadt.de]

Prototyping 5G using LabView SDR Platform (AA1)

We developed a Software Defined Radio (SDR)-based platform for experimental research on 5G Device-to-Device communication. This prototype have some limitations (e.g., compatibility with commercial devices and number of supported users) that can be alleviated by migrating it to USRP FlexRio and LabVIEW communications design suit.



Your tasks:

- Become familiar with D2D communications.
- Become familiar with LTE and WiFi reference design code in Labview.
- Migrate the current code to the new testbed.
- Test and evaluate the performance of new testbed.

Type: Lab/Project for 3-4 students

Prerequisites:

Basic knowledge of LTE and WiFi standard, LabVIEW.

Contact:

Arash Asadi (arash.asadi@seemoo.de)

Topic Overview

MW1: Replacement of BGP Routers (2 - 3 Students)

MW2: STEAN OS (2 - 4 Students)

MW3: Greetings from the Network (1 Student)

MW4: Virtualizing the Testbed (2 - 4 Students)

JC1: Android Signal Analyzer (1 - 4 Students)

JC2: Visible Light Communication (2 - 4 Students)

DS*2: Simulating Highly Directional 60 GHz Mesh Networks
(2 - 4 Students)

MiS*1: WMN Inspect: Implement a network traffic and
topology analyzer (2 - 4 Students)

MiS2: Implementing 'ping' and 'traceroute' for Castor
(1 - 2 Students)

MiS3: Replacing Serval's Mesh Routing with a Secure
Protocol (2 - 3 Students)

MiS4: Click2iOS (1 - 2 Students)

MaS1: Implementing 802.11 on USRPs (2 - 4 Students)

MaS2: Develop the SEEMOO software-defined radio
(4 - 8 Students)

MaS3: WiFi Firmware Hacking (2 - 4 Students)

DY1: Implementing and Improving Constructive
Interference with SDR (1 - 2 Students)

DY2: Data Collection with TSCH (2 - 3 Students)

AB1: Implementing aDTN on Click (2 - 4 Students)

FA1: Framework for supporting multiple wireless
technologies on Android (2 Students)

FA2: Analysing energy consumption on android
(1 - 2 Students)

RK*1: Time-Synchronized WiFi for WARP (2 - 4 Students)

RK2: Decoding Wireless Collisions (2 - 3 Students)

RK3: GPS-based 3D Compass (2 - 4 Students)

RK*4: ns-3 Channel Model for WMNs (2 - 4 Students)

LA*1: High-Level Testbed Framework (2 - 4 Students)

MF1: What is up with your WhatsApp? (2 Students)

AA1: Prototyping 5G using LabView SDR Platform
(3 - 4 Students)