

MASARYK UNIVERSITY  
FACULTY OF INFORMATICS



# **Edu-hoc: Experimental and educational platform for wireless ad-hoc networking**

MASTER'S THESIS

**Lukáš Němec**

Brno, Fall 2016

MASARYK UNIVERSITY  
FACULTY OF INFORMATICS



# **Edu-hoc: Experimental and educational platform for wireless ad-hoc networking**

MASTER'S THESIS

**Lukáš Němec**

Brno, Fall 2016

*Replace this page with a copy of the official signed thesis assignment and the copy of the Statement of an Author.*

## **Declaration**

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Lukáš Němec

**Advisor:** RNDr. Petr Švenda, Ph.D.

## **Acknowledgement**

TODO thanks

# Abstract

TODO abstract

## **Keywords**

keyword1, keyword2, ...

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Problem analysis (testbed, not general WSN)</b>	<b>2</b>
2.1	<i>Creating WSN network</i> . . . . .	2
2.2	<i>Possible challenges</i> . . . . .	2
<b>3</b>	<b>TESTBED deployment</b>	<b>3</b>
3.1	<i>Network design</i> . . . . .	3
3.2	<i>JeeTool (mass managment and communication)</i> . . . . .	3
3.3	<i>HW (Arduino, JeeNodes, RF12B radio ...)</i> . . . . .	3
<b>4</b>	<b>Research use</b>	<b>4</b>
4.1	<i>Keys from radio signal</i> . . . . .	4
4.1.1	Quantization principle (bits from signal strength)	4
4.1.2	RSSI version . . . . .	4
4.1.3	CSI (channel state) version . . . . .	6
4.2	<i>Cooperative jamming (can it improve our situation?)</i> . . . . .	6
4.3	<i>Performance Evaluation (results from experiments)</i> . . . . .	6
4.3.1	Enthropy of data . . . . .	6
4.3.2	Speed (bits of key per time) . . . . .	6
4.3.3	Possible errors . . . . .	6
4.4	<i>Discussion, is it achievable and under what conditions?</i> . . . . .	6
<b>5</b>	<b>Education use</b>	<b>7</b>
5.1	<i>motivation for educational WSN network</i> . . . . .	7
5.2	<i>Scenario approach (attack and repair) + iterative higher dif-</i> <i>ficulty</i> . . . . .	7
5.2.1	scenarios . . . . .	8
5.3	<i>Evaluation principle</i> . . . . .	9
5.4	<i>Web interface and auto run</i> . . . . .	9
5.5	<i>PA197 use and results</i> . . . . .	9
<b>6</b>	<b>Summary</b>	<b>10</b>
<b>A</b>	<b>An appendix</b>	<b>11</b>



# **1 Introduction**

## **2 Problem analysis (testbed, not general WSN)**

### **2.1 Creating WSN network**

### **2.2 Possible challenges**

## **3 TESTBED deployment**

### **3.1 Network design**

### **3.2 JeeTool (mass management and communication)**

### **3.3 HW (Arduino, JeeNodes, RF12B radio ...)**

## 4 Research use

### 4.1 Keys from radio signal

#### 4.1.1 Quantization principle (bits from signal strength)

Quantization enables to extract bits from individual values of signal strength. There are many different approaches to this problem two main approaches here are: lossless quantization and contrary to that we have lossy quantization.

Main difference between these is number of generated bits per original signal strength measurements, while lossless quantization produces bit value from every measurement of signal strength, which is usefull for high performance demands, but it requires guaranteed varinace in the radio channel (e.g. the nodes are constantly moving, or the nevironment is changing) during the key establishment phase. Otherwise the resulting keys could possibly be very weak.

Lossy quantization on the other hand does not have guaranteed output lenght per number of measured values, which can lead to very limited length of output. However this kind of quantization is expected to have better results in static enviroments because its nature is to drop such bits, that fail to differ from others.

Since our network is static and without any moving nodes, we implemented lossy quantizer algorithm designed by Mathur et.al., which shoved promissing results for the of the shelf wireless devices simmlar to ours, and also contained experimental results from several different scenarios, where some of these were comparable to our conditions.

#### 4.1.2 RSSI version

Quantization princeple designed by Mathur et.al. works as follows:

1. both nodes send  $n$  messages to each other in alternating pattern, both nodes send counter value inside these messages, which is used to synchronize messages on individual nodes. For every one of these messages signal strength is measured upon repection.

2. when  $n$  messages have been successfully exchanged, both nodes can proceed to the computational part.
3. both nodes calculate mean  $m$  and standard deviation  $sd$  for signal strength values of all received messages.
4. both nodes calculate  $q^+$  and  $q^-$  values, which are upper and lower quantizer bin boundaries, as follows:

$$q^+ = m + \alpha \cdot sd$$

$$q^- = m - \alpha \cdot sd$$

5. every signal strength measurement is then processed and it is rejected, if it lies within  $q^+$  and  $q^-$  boundaries, values above this range are assigned with bit value of one, values below are assigned with bit value of 0.
6. nodes then synchronize their measurement by exchanging counter values associated with those messages, where signal strength measurements were assigned either one or zero bit values.
7. those counter values that match on both nodes are expected to be excursions in the same direction and are used in the final outcome.

**4.1.3 CSI (channel state) version**

**4.2 Cooperative jamming (can it improve our situation?)**

**4.3 Performance Evaluation (results from experiments)**

**4.3.1 Entropy of data**

**4.3.2 Speed (bits of key per time)**

**4.3.3 Possible errors**

**4.4 Discussion, is it achievable and under what conditions?**

## **5 Education use**

### **5.1 motivation for educational WSN network**

The current state of the art WSN devices usually uses specialized hardware and software in order to achieve the best performance available. This, unfortunately, is not the ideal prerequisite for an easy to learn matter. In fact, most of WSN devices have rather complicated setup and are quite challenging for novices.

Because of such discouragement, it is difficult to teach how to work with WSN's; few hours (at least) are usually required to explain the basics, which is reasonable for research project or something similar, but for class exercise, this would turn out to be not the most effective use of time, if it would be achievable at all. And we have not yet mentioned more advanced topics in this area, such as common techniques for encryption or message authentication.

Issue of this nature can be solved in various ways, in case of Edu-hoc we decided to sacrifice performance 3.3; which is not that much important for network with educational purpose. On the other hand, using hardware that is really easy to comprehend and use is of a great benefit here. Also having less powerful, but relatively cheap devices (in range of \$30 rather than \$100 or more) gives the opportunity to lend each the students one of the devices, so that they can try the basics on their own, and also use this device for interactions with the network.

### **5.2 Scenario approach (attack and repair) + iterative higher difficulty**

In order to make learning more enjoyable experience and also to add some challenging part to the learning process, we decided to make Edu-hoc scenario based; each scenario being composed of two distinct parts: first part in the role of an attacker (both enjoyable and educational) and the second part as a code reviewer or developer (primarily educational).

**attacker part of a scenario** Students are presented with network application which has known, or easily detectable, vulnerability. Task is to take advantage of such vulnerability and exploit it using own nodes and carefully executed interactions with the network. How successful these efforts were can be easily evaluated by percentage difference from the expected traffic of the network or by presenting learned secret in the submission of the solution. Exact methods of evaluation are described in 5.3.

**reviewer part of a scenario** Important part of the educational process is not only to find mistakes, but also to be able to correct them. The task in this part of scenario is then to take current source code of application, which has been deployed on the network during the attacker part 5.2, correctly identify which mistakes were made and propose changes in the code, which would make the application secure against such kind of an attack. Exact evaluation methods are again described in 5.3.

### 5.2.1 scenarios

#### - 5.2.5 individual scenarios

##### 1st scenario - Eavesdropping

First scenario can be considered really simple and it is by design. Since this scenario is the first encounter with Edu-hoc, we opted for passive attacker approach, without any actual interaction with the network, just listening for any traffic.

Eavesdropping is also the first thing one would do, if one would be about to attack some network, because it does not compromise the presence of an attacker and intercepted messages may provide many useful informations. This is another important reason why this task was selected as a first one, not only it is rather easy to do, but it provides vital information for the rest of the exercises (e.g. which nodes are present in the network, which frequency is used and what are the settings of the radio)



**scenario setting** Each node in the network sends messages, after each message there is a delay of 1000 ms, therefore any receiving node has time to process the transmitted messages. All messages are send as a broadcast and nodes within the network do not receive nor process them, since it is not needed for succesfull scenario run. If this would be the case, transmission rates would have been updated acordingly and scenario would be much more likely to fail on its own, because of single node malfunction, while with current settings, network can operate without any problems even is several of the nodes would fail.

**attack principle** Attack is very simple, attacker node only has to listen for any traffic present in the network. For best results application has to be able to process as many messages as possible, however any performance changes will affect the end result only slightly (final percentage will be better, but the bottle neck is in the processor of the radio module, so better optimalsed application will increase the end result only by approximately 10-20%)

**application and securing it** Main issue with the application is un-encrypted broadcast. This can be fixed quite easily, just by addition of simple encryption and for simplicity using common shared key. We have to be avare of the fact, that attacker might be able to collect some of the nodes and thus learn such key and compromise it and there are techniques which deal with this problem However, in our scenario this is sufficient solution, because it is adequate measure against eavesdropping.

### 5.3 Evaluation principle

### 5.4 Web interface and auto run

### 5.5 PA197 use and results

## 6 Summary

## **A An appendix**