Team project
"Software for self-testing of the Telecommunication network of University of Freiburg"

Arda Akcay
Tri Atmoko
Refik Hadžialić

October 5, 2011

Albert-Ludwigs-Universität Freiburg
Lehrstuhl für Komunikationsysteme
Prof. Dr. Gerhard Schneider

Supervisors:
Konrad Meier
Denis Wehrle

Sommersemester 2011

Contents

1 Introduction and Motivation

In the following report, the authors will try to give you a brief insight into our team project. The goal of our project was to develop a mechanism for automatic testing of our University Telecommunication network. The Telecommunication network of University of Freiburg consists of: our own internal GSM and telephone network systems; GSM redirecting device (if one initiates a call to one of the four external GSM networks, it redirects the calls to: T-mobile, 02, Vodaphone or E-Plus); a SIP gateway for landline calls inside of Germany (sipgate.de) and international calls. Since we did not have access to internal servers, our strategy was to exploit the existing systems and infer the results out of our findings. Before we had started working on our project, we had to analyze the overall network to come up with test cases that contain the highest information content. The next step in our procedure was to implement our ideas into a working piece of software. Gradually we implemented a bit-by-bit of the final software. Every single step was accompanied by testing and validation procedures. At the end we connected all the "black-boxes" into one big piece of software. We have fulfilled our requests and goals and made a fully working and operable test software. Despite developing a working software, all the way along we thought about the simplicity of the usage of the software. In the following chapters we will describe in more detail our approach and how each subsystem works.

2 Software concept

3 Introduction

3.1 Usage

4 Design

5 Protocol

6 Encryption of data

```
1  import subprocess
2  import string
3
4  class Ping:
5
6          def __init__(self, pingAddress):
7                  self.pingAddress = pingAddress
8
9          def ping(self,numberTries):
10                 tried = 1
11                 while numberTries >= tried:
12                         tried += 1
13                         #the parameter c 1 means only one ping to be sent, parameter W 3 means
                               how many seconds the time out should be, 3 seconds
14                         ping_cmd = subprocess.Popen(['ping', self.pingAddress, '-c', '1', '-W', '2'],
                               stdout=subprocess.PIPE, stderr=subprocess.STDOUT).communicate()
                               [0]
15 b
16                         pingAlive = int(string.find(ping_cmd, '1 received'))
17                         unknownHost = int(string.find(ping_cmd, 'unknown host'))
```

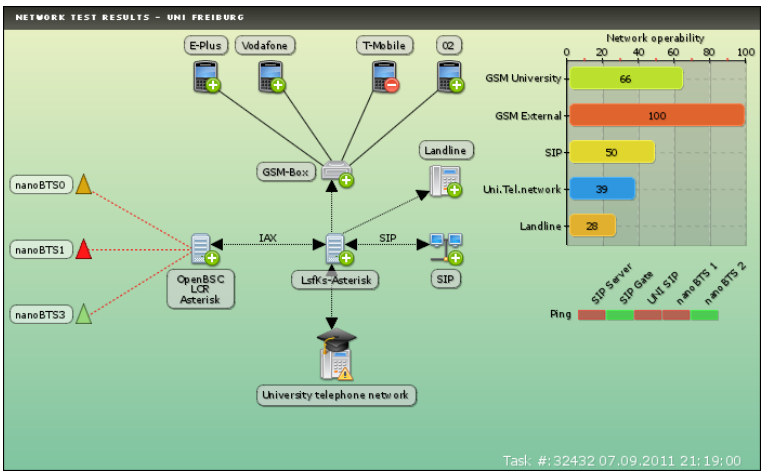

Figure 1: Result image showing working, defected and not tested subsystems

7 Web page

8 Conclusion

References

[1] H. Simpson, *Proof of the Riemann Hypothesis*, preprint (2003), available at `http://www.math.drofnats.edu/riemann.ps`.