



**KTH Computer Science  
and Communication**

# **Configuration and device identification on network gateways**

Configuration and device identification on network gateways

SIMON KERS

Bachelor's Thesis at STH  
Supervisor: Micael Lundvall  
Examiner: Ibrahim Orhan



# Abstract

To set up port forwarding rules in their network gateways, certain technical skills are required from the user. These assumptions in the gateway software stack, can lead to an increase in support calls to the network operators and resellers of equipment, as well as a worsened user experience. Other issues include faulty configuration, leaving the network vulnerable to attacks.

We designed a graphical user interface as an OpenWrt module and implemented a wrapper for a network scanner to detect devices and applications on the local network. This enables end-users to avoid looking up forwarding rules for ports and protocols to configure port forwarding rules, basing their decisions on data from the network scanner.

The implementation will reduce support costs for the service operators and improve user experience.



# Referat

## Konfigurering och enhetsidentifiering på nätverksgateways



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Current situation . . . . .	3
2.2	Software suite . . . . .	4
2.2.1	OpenWrt . . . . .	4
2.2.2	OPKG . . . . .	4
2.2.3	Inteno Open Platform System . . . . .	4
2.2.4	Lua Configuration Interface . . . . .	4
<b>3</b>	<b>Problem</b>	<b>7</b>
3.1	User experience . . . . .	7
3.2	Support costs . . . . .	7
<b>4</b>	<b>Design</b>	<b>9</b>
4.1	Human-computer interaction . . . . .	9
4.2	Forwarding page . . . . .	9
<b>5</b>	<b>Implementation</b>	<b>11</b>
<b>6</b>	<b>Results</b>	<b>13</b>
6.1	Performance . . . . .	13
<b>7</b>	<b>Conclusions</b>	<b>15</b>
7.1	Further development . . . . .	15
	<b>Appendices</b>	<b>15</b>
	<b>Bibliography</b>	<b>17</b>





# Chapter 1

## Introduction

Inteno Broadband Technology is a company that supplies customer premises equipment for internet service providers. Their headquarters and research and development center is located in Stockholm, Sweden. Inteno Open Systems Platform is a Linux-based open source platform running on customer premises equipment. It is based on the OpenWrt distribution which targets embedded devices, specifically network gateways. [1]

The technical support departments of partners and resellers of Intenos customer premises equipment, are looking to reduce support costs and improve customer experience. Support issues creates costs for the business and by reducing the number of support tickets and their processing time, these costs can be reduced.

By simplifying configuration through abstracting common tasks for the end-user, the number of support calls can be reduced. Using automatic device identification and automating common tasks such as port forwarding, support costs can be reduced and end-user satisfaction is higher. Many common support issues could be automated by the software running on the customer premises equipment and by effective communication with the end-user through the user interface.

The OpenWrt distribution provides a complete platform for compiling and deploying a gateway firmware image. By building a extensible library of presets for common port forwarding rules and developing a simple selection dialog, the amount of calls can be lowered while increasing customer satisfaction.



## Chapter 2

# Background

Inteno produces hardware for customer premises equipment like network gateways and their direct customers are network operators and internet service providers. The research and development department at Inteno works on improving the platform, adding value to the end users and the operators.

There are simple ways in which to improve the user experience, developers of network gateway software often implement a set of presets of port forwarding rules for common applications. The interface presents the user with a list of applications in clear text and lets the user select an IP address, for which the forwarding rules should apply.

Alternative solutions to simplifying port forwarding include using standalone applications which run on a PC, connected to the local network. These applications has internal lists of port forwarding rules for common applications and devices, which is then applied for a specific IP address on the local network. [3]

To test the newly applied configurations, web-based or locally run port scanners can be used. They will scan the users external IP address for open ports and present which are open, this does not guarantee that the packets are routed to the correct internal address.

### 2.1 Current situation

The default settings page for port forwarding is currently located under the *Firewall* tab, the forwarding procedure involves looking up ports for the specific device or unit. These set of rules sometimes involve several ports and protocols, increasing the possibility for misstep and faulty configuration by the end user. Such tasks are well suited for automation by the software suite, especially for applications and devices which require several port forwarding rules, automatic some of these steps will save time and bring overall value to the user experience.

## 2.2 Software suite

### 2.2.1 OpenWrt

OpenWrt is a free and open-source GNU/Linux distribution, targeting embedded devices, specifically wireless routers, but can run on almost any set of hardware. Cross-compilation is enabled by OpenWrt Buildroot, which compiles the C code using uClibc, a lightweight C library focusing on embedded Linux systems. It intends to be a meta distribution and offers developers a framework on which to base their firmware on.

OpenWrt is generally compiled and linked using gcc and binutils, with the help of Makefiles and patches for the various gcc versions and target platforms. Allowing end users as well as service operators and hardware manufacturers to compile the firmware. It offers the BusyBox set of barebones UNIX tools, enabling advanced users to fully interact with their Linux system and providing developers with a familiar platform for debugging and testing their product. [2]

### 2.2.2 OPKG

The package management system used in OpenWrt is *OPKG*. It is based off the discontinued *ipkg* and operates similar to *APT* and *dpkg* of Debian-based distributions. There are currently over 2000 OPKG packages available for OpenWrt.

The OpenWrt system and its packages are built using *GNU Autoconf*.

### 2.2.3 Inteno Open Platform System

For Customer Premises Equipment<sup>1</sup> like wireless gateways, Inteno Open Platform System offers an open-source Linux distribution based on OpenWrt. It uses the OpenWrt build system including cross-compilation toolchain to ensure compatibility with the ecosystem and upstream.

### 2.2.4 Lua Configuration Interface

*LuCI* is an suite of programs and libraries for extending OpenWrt using the Lua programming language. It originated in the OpenWrt project but has since grown and is now it's own project.

The themes are accessed from the directory:

```
root@Inteno:/usr/lib/lua/luci/view/themes/
```

Rules for port forwarding are read from:

```
/etc/config/firewall
```

---

<sup>1</sup>commonly abbreviated as CPE

## 2.2. SOFTWARE SUITE

A port forwarding rule which forwards external HTTP traffic over port 80 to the local IP 192.168.1.214, looks like:

```
config redirect
    option target 'DNAT'
    option src 'wan'
    option dest 'lan'
    option proto 'tcp'
    option src_dport '80'
    option dest_ip '192.168.1.214'
    option dest_port '80'
    option name 'Web server'
```

The presentation markup for the current port forwarding page in the LuCI backend on the Gateway, is defined in the file:

```
luci-inteno/applications/luci-firewall/luasrc/view/firewall/cbi_addforward.htm

libs/core/luasrc/model/firewall.lua :555
```

in the functions:

```
function redirect.*
```



## Chapter 3

# Problem

### **3.1 User experience**

Customers and operators of Inteno CPE have expressed concern about support costs for port forwarding and configuration in general. The

### **3.2 Support costs**





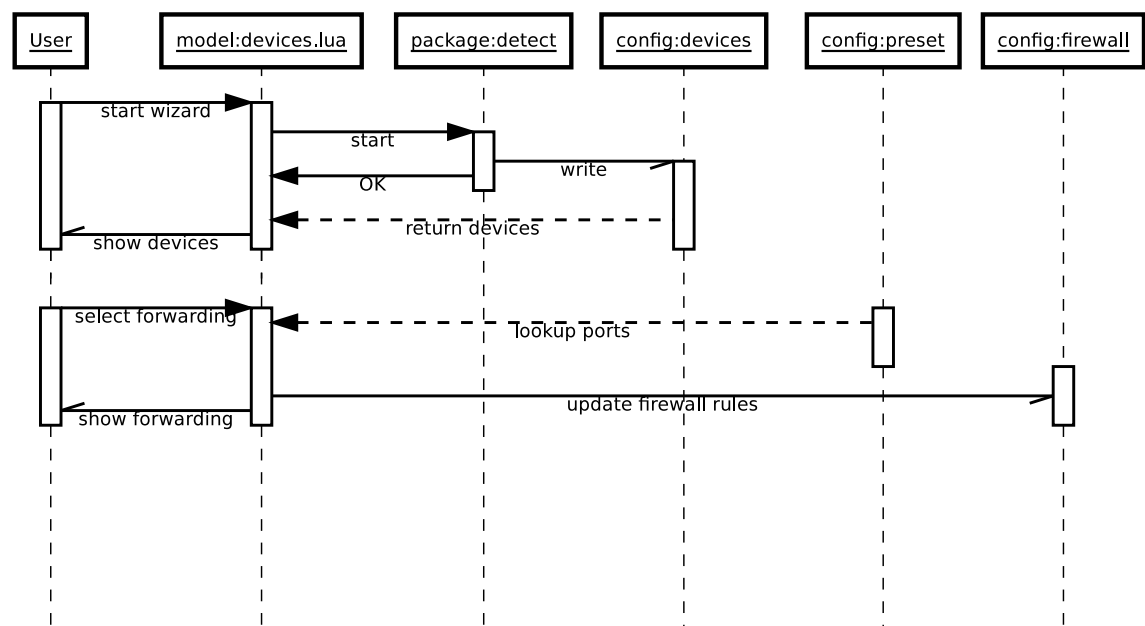
## Chapter 4

# Design

### 4.1 Human-computer interaction

### 4.2 Forwarding page

See figure 4.1 on page 9.



**Figure 4.1.** Sequential diagram of applying port forwarding rules



## Chapter 5

# Implementation

The implementation consists of three general parts that work together in delivering easier forwarding rules configuration. As shown in figure 4.1, the three configuration files that together with user input, are used as sources for the final redirection rule in the firewall configuration file. The device detection updates the configuration file *devices* with newly discovered devices, this does not include applications running on computers or game specific forwarding rules, running on gaming consoles, this requires user intervention.

Using the *Unified Configuration System*, that is included in the OpenWrt distribution, all the basic commands for configuring the firewall rules were prototyped and explored.



## **Chapter 6**

# **Results**

### **6.1 Performance**



## **Chapter 7**

# **Conclusions**

### **7.1 Further development**





# Bibliography

- [1] New business possibilities with Open Source software. [http://www.inteno.se/Portals/0/IntenoFiles/ProductDocs/241/689/iopsys\\_white\\_paper.pdf\\_20121015135755.pdf](http://www.inteno.se/Portals/0/IntenoFiles/ProductDocs/241/689/iopsys_white_paper.pdf_20121015135755.pdf). Accessed: 2013-04-29.
- [2] OpenWrt structure and design. [http://wiki.confine-project.eu/\\_media/soft:openwrt-talk-2012-06-01.pdf](http://wiki.confine-project.eu/_media/soft:openwrt-talk-2012-06-01.pdf). Accessed: 2013-04-29.
- [3] Port Forward homepage. <http://portforward.com/>. Accessed: 2013-05-14.