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

Abstract in English.

Referat

Konfigurering och enhetsidentifiering på nätverksgateways

Innehåller en svensk sammanfattning av rapportens innehåll på 10-15 rader samt nyckelord som beskriver innehållet (upp till 10 stycken). Sammmanfattningen bör kortfattat redogöra för frågeställningen/problemmet, metoden och svaret/resultateet så precist som möjligt.

Contents

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 duration, 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 effectively communication with the end-user through the user interface.

The OpenWrt distribution provides a complete platform for compiling and deploying a gateway 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

2.1 Software suite

2.1.1 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]

2.1.2 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.

2.1.3 Inteno Open Platform System

For Customer Premises Equipment[1] 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.

[1] commonly abbreviated as CPE

2.1.4 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
```

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

3.2 Support time

Chapter 4

Related work

4.1 Human-computer interaction

Chapter 5

Implementation

5.1 Design

See figure 5.1 on page 9.

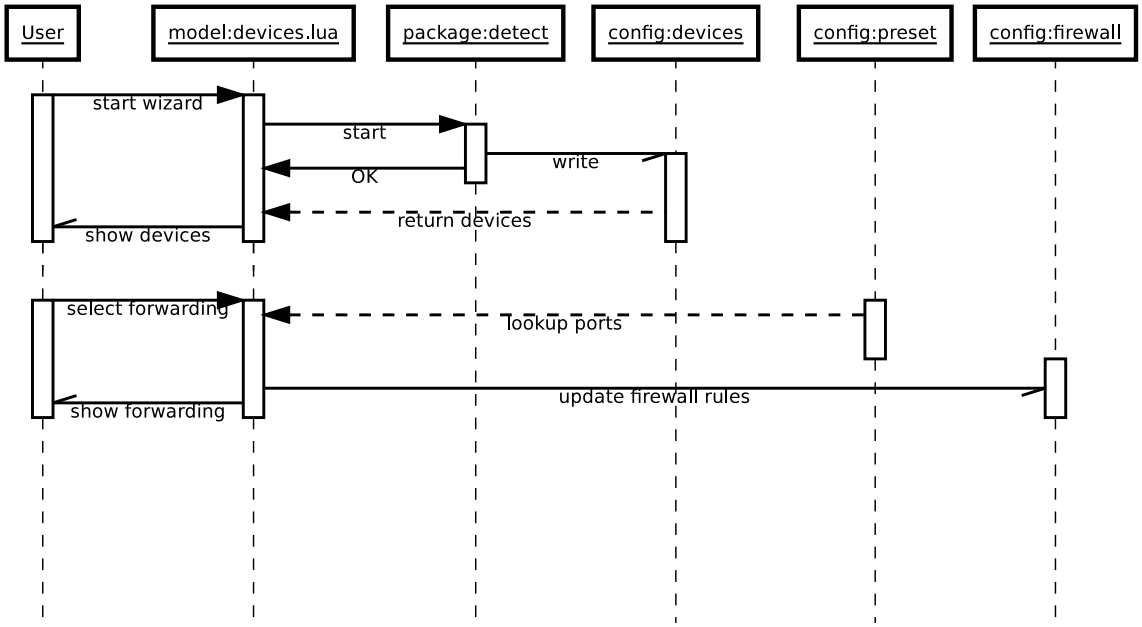

Figure 5.1. Sequential diagram of applying port forwarding rules

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. Accessed: 2013-04-29.

[2] OpenWrt structure and design. http://wiki.confine-project.eu/_media/soft:openwrt-talk-2012-06-01.pdf. Accessed: 2013-04-29.