Liberouter: a PC-based IPv6 router

Ladislav Lhotka
<lhotka@cesnet.cz>

Jiří Novotný
<novotny@ics.muni.cz>
Introduction

• Routers and routing – complex technology, interesting algorithms inside the black (blue, violet) boxes
• Liberouter: let’s do it ourselves, as a truly open research project (open-source licenses)
• Flexible platform for network research, especially in IPv6
• Competitive solution for the network edge
• Special functions: encryption, monitoring, traffic generator
• Commodity PC with additional hardware, later as embedded systems
• Support from CESNET and 6NET
Software PC router

• In CESNET networks since early 1990s
• Operating systems: KA9Q, PC Route, *BSD, Linux
• Routing software: GateD, Zebra
• Excellent experience, still in use in some MANs, national IPv6 backbone, m6bone
• Limited forwarding performance (PCI and memory bus throughput, interrupt handling)
• Inconsistent configuration through init scripts and configuration files
GE forwarding performance

PC/Linux (PCI 64/66)  
Cisco 7500

(a) 1500 B  
(b) 82 B  
(c) Mix

Load [Mbit/s]  
Time [s]

Load [Mbit/s]  
Time [s]

Load [Mbit/s]  
Time [s]
• Dual-stack router with hardware acceleration:
  ▶ Commodity PC with NetBSD or Linux
  ▶ FPGA-based board: data plane in HW; control plane remains in SW
  ▶ Throughput about 10 Gbps
  ▶ Daughter interface cards (GE, 10GE, later POS)

• Powerful and user-friendly configuration system
  ▶ XML-based configuration repository
  ▶ Bidirectional translation between XML and native configurations (Unix files, IOS, JUNOS), later XML-RPC
COMBO6 architecture

No processor, combination of programmable hardware and standard integrated circuits

- Xilinx FPGA (Virtex II 3000–8000), CPLD
- CAM, RAM, DRAM, PCI interface, power supply
- Exchangeable interface cards
- Test/extension connector
Firmware

- Modular design
- VHDL – standard development approach with simulation, requires expensive development environment
- Concept of “nanoprocesors” makes simple reprogramming possible
- Nanoprocesors are also simulated
- Hardware/software co-design: algorithms are first implemented in software and their time-critical parts are gradually moved to hardware
- Support from formal verification group
Packet processing

**Diagram:**
- **PCI**
- **LUP**
- **REP**
- **QUE**
- **HFE**
- **DRAM**
- **OPE**

**Nodes:**
- **in**
- **out**

**Arrows:**
- Unif. header
- Payload
System software

• Development for NetBSD and Linux
• COMBO6 driver presents the card to the system as a standard 4-port Ethernet card
• combo6 daemon with hooks in the OS: necessary changes are propagated into the COMBO6 card
• Configuration can be done mostly by standard Unix utilities (ifconfig, route, …)
• Standard routing daemons can be used
• tcpdump support
System software (cont.)
Current status

• 4 COMBO6 cards manufactured and tested (~€3.500)
• Interface card 4×GE (copper)
• Low-level driver for NetBSD and Linux
• Development tools – *comboctl*, TCL scriptable
• Intensive work on the firmware and software
Netopeer configuration system

• Powerful platform-independent environment for configuring large heterogeneous networks
• Central configuration repository with version control
• Internal data format based on XML
• Netconf group in IETF: communication protocol (XML-RPC), not data contents
• Netopeer development concentrates on configuration data and their transformations
Netopeer architecture

XML configuration data

- CLI
- WWW
- SNMP
- legacy
- metaconfig
- conf. files
- XML-RPC
- SNMP
Current status

- Preliminary XML schema (DTD): interface configuration (incl. VLANs and tunnels), packet and route filtering, static routes, RIP and RIPng.
- Front-ends: Cisco IOS, JUNOS, web
- Back-ends (XSLT): Cisco IOS, Linux/NetBSD
Further information

• Project site – http://www.liberouter.org
• All project files publicly available from CVS
• Mailing lists
• Project team has already 45 members, but contributors are still very welcome!