XORP
Extensible Open Router Platform

Niklas Steinleitner

Telematics Group

Advanced Topics in Computer Networking

Institute for Informatics

SS 2005

University of Göttingen
Table of Content

- Motivation
- Alternatives
- XORP
  - Main Challenges
  - Design Overview
  - Important modules
  - Security Framework
  - Performance Evaluation
- Summary
Motivation

- Network Researchers’ Problems
  - router software market is closed
  - routers run only vendor’s software, commercial interests
  - router platforms and API's generally not open systems
  - problems when deploying software in router
    - for experimental or pilot deployment in real networks
  - vendor’s have robustness and security as main goals
    - not provide API’s that allow third-party extension
  - router vendors must implement new protocols
    - unlikely to invest resources into a feature without existing results

⇒ Network Researchers need capabilities to evaluate new protocols
Alternatives

- Vendors open their internal API’s for experimental use
- Network Simulators
  - NS-2
  - Opnet Modeler
- Network Testbeds
  - DARTnet
  - CAIRN
- Open Router Systems
  - only real-world experimentation can completely evaluate an approach
  - GateD
  - Zebra
  - MRTd
  - XORP
XORP: Extensible Open Router Platform

- Open Source Router Platform
- ICSI Berkeley, California
- BSD-style license
- Exists for Linux, FreeBSD, as a LiveCD
- as research and production platform
  - Low Cost Router on conventional PCs
- implements BGP, OSPF, RIP, …
XORP Main Challenges

- **Features**
  - routing protocols, management interfaces, queue management

- **Extensibility**
  - routing protocols, forwarding engine, API

- **Performance**
  - not designed for core routers, nevertheless performance is important

- **Robustness**
  - according to extensibility and performance
XORP Design Overview

- event-driven (avoiding delays of timer-based designs)
- router functionality separated into many UNIX processes (robustness)
- IPC mechanism lets modules communicate with each other, independent of the fact that modules are part of same process or even on same machine
- allows untrusted processes to run sandboxed
- XORP is divided into two subsystems
  - higher-level (user-level)
  - lower-level
Higher Level (User-Level)

- routing protocols
- routing information base
- supports processes
- multi-process architecture, one process per routing protocol or management, configuration, coordination
- for extensibility inter process communication mechanism called Finder
Lower Level

- provides APIs for higher levels
- manages forwarding path
- alternative forwarding paths
  - Click modular router
    - a modular, extensible toolkit for packet processing in conventional PCs
  - conventional FreeBSD lower level
  - other forwarding paths
    - FreeBSD with different extensions
    - new extensible forwarding path
Four Core Processes

- Router Manager
- Routing Information Base (RIB)
- Forwarding Engine Abstraction (FEA)
- IPC Finder
Router Manager

- manages the router as a whole
- holds router configuration
- starts, configures, and stops
  - protocols
  - other router functionality
- restarts failed processes if necessary
- hides the router's internal structure
- management interfaces
Routing Information Base (RIB)

- receives routes from routing processes
- decides which routes propagate into the forwarding path
- redistributes to other routing processes
- critical for the correct functionality of a router
- normally not extended
- should ideally be general enough to cope with all routing protocols
Forwarding Engine Abstraction (FEA)

- Provides a stable API for communication with forwarding engines
- Abstracts the details of how the forwarding path is implemented
- Manages the networking interfaces and forwarding table
- Provides information to routing processes about
  - Interfaces properties
  - Occurring events on interfaces
- With the Finder, processes can bypass the FEA if required
IPC Finder

- allows communication both between XORP processes and routing applications not built using XORP framework
- uses multiple transport transparently
  - intra-process calls
  - host-local IPC
  - networking communication
- discovers how to make a IPC call and advised application
- proceeds via naturally scriptable base called XORP Resource Locator (XRL)
- XRLs human-readable, like URLs
Security Framework

- critical aspect by a extensible platform
- ideally no damage to the router by an experimental module
- memory protection
  - provided by multi-process architecture
- sandboxes, no access to important part of the filesystem
  - configuration information centralized in router manager
  - no process needs access to the filesystem
- performs privileged network operations needs root access
  - FEA is used as a relay for all network access using XRLs
- leaving XRLs as remaining damage factor
  - local circumvention and bypassing the finder are prevented by an 16-byte random key in the registered method name of XRLs
- several plans for extending XORP security
  - unique secret
  - run processes in different virtual machines
XRL Performance Evaluation

- XRL IPC mechanism might become a bottleneck
- evaluated three communication transport mechanism
  - TCP, UDP, intra-process via XRL

![Graph showing XRL performance for various communication families]
Event-Driven Performance Evaluation (I)

- argued that event-driven route processing is faster than the traditional route scanning approach
- introduced 255 BGP routes every second from BGP peer and recorded the time that the route appeared at another BGP peer
Event-Driven Performance Evaluation (II)

- example records the time at which the route “10.0.1.0/24” has been added
- into system with empty routing table and a system with a full BGP backbone feed of 146515 routes
- a new route every two seconds and removed after one second
Summary

- primarily intended to provide network researchers a Low Cost Router
- XORP achieves main challenges

  - Features
    - BGP4+, OSPF, RIPv2, PIM-SM, IGMPv3/MLD, (IS-IS)
    - both IPv4, IPv6
    - command line interface, SNMP
    - mostly resembling existing code

  - Extensibility
    - open interfaces are the key to extensibility
    - open inter-process interfaces and the XRLs forms the cornerstone for XORP's extensibility

  - Performance
    - for a PC-based hardware platform scales well
    - lower level performance depends on forwarding path selection

  - Robustness
    - processes are protected from each other
    - router manager can restart crashed processes
    - security framework provides robustness in higher level

- good extensibility is ensured
Thanks for your attention!
Questions?