Extensible Networking in Reconfigurable Hardware
Invited Talk at Georgia Tech's Center for Experimental Research in Computer Systems
February 20, 2003
John W. Lockwood
Applied Research Laboratory
lockwood@arl.wustl.edu
http://www.arl.wustl.edu/~lockwood/

Fundamental Changes

- **Existing (Old School) Internet Technologies**
  - Shortest hop routing
  - Best effort packet delivery
  - Bit transport service
  - Current Technologies
    - Microprocessors (slow)
    - ASICs (static)
  - Current Problems
    - Network routers have static functionality
    - Packet filtering rules manually deployed into firewalls
    - Services manually deployed into network
    - Network becomes unstable when attacked by Internet worms and DDoS outbreaks

- **Extensible Network (New School) Services**
  - Best path routing
  - Guaranteed Quality of Service
  - Application-Level services
  - New Technologies
    - FPGAs
    - Network Processors
  - New Possibilities
    - Network routers with dynamic feature set
    - Packet filtering rules automatically distributed to packet filtering nodes
    - Services automatically deployed into network
    - Robust for all possible traffic patterns
**Configuration of Network Packet Processor**

- **Packet processing hardware performs:**
  - Packet classification
  - Packet forwarding
  - Address Translation
  - Data modification
  - Packet buffering
  - Active Networking (Application-level data processing)

**Configuration of Internet Router**

- **Additionally, Router interface performs:**
  - Internet route lookup
  - Traffic policing and shaping
Practical Platform for Extensible Hardware

- Suitable platform for shared development
  - Open platform

- Well-defined hardware interface
  - Suitable for System-on-Chip (SOC) Implementation

- Complete design methodology
  - Automatic flow from design to implementation

- Physical Implementation tools
  - Allow synthesis, place, and route with existing tools

- System management tools
  - Link static and dynamic functionality

Field Programmable Port Extender (FPX)

- **RAD**
  - Large Xilinx FPGA
  - Attaches to SRAM and SDRAM
  - Reprogrammable over network
  - Provides two user-defined Module Interfaces

- **NID**
  - Provides Utopia Interfaces between switch & line card
  - Forwards cells to RAD
  - Programs RAD

![Diagram of RAD and NID connections](image-url)
Port Processing at edge of Gigabit Switch

- **Original Network Switch**
  - Line card connects to Gigabit switch backplane

- **FPX-Enhanced Router**
  - Line card connects to Gigabit switch backplane

Photograph of the FPX
Stacked FPX / SPC / Line card

Washington University Gigabit Switch with FPX Devices
Switch, FPGA, & Processor Configuration

Applications for Extensible Networks:

Network Security
Dynamic Hardware Plugin contains:

- **Processing Module**
  - Processes Data passing through the module

- **Protocol Wrappers**
  - Segment and reassemble Internet packets
  - Compute packet headers, lengths, and checksums
  - Provide reliable stream of data to processing module

- **Interfaces**
  - Read and write packets to network
  - Buffer data in off-ship SRAM and SDRAM

Content-Aware Firewall on a Chip

- Layered Protocol Wrappers
  - Off-Chip Synchronous Random Access Memory (SDRAM)
  - Off-Chip Static Random Access Memory (SRAM)

- Identify packets based on content-based match (regex)
- SDRAM Controller
  - SDRAM Free List Manager
  - SDRAM Controller
    - SDRAM Free pointers
  - SRAM Controller
    - SRAM Free pointers
- CAM-based Firewall
  - Match vector
  - Flow from CAM
- Flow Buffer
  - Head Pointers
  - Tail Pointers
- Scheduler (RR, DRR, 3DQ)

(Implemented on the RAD on the FPX, using a VirtexE 2000 FPGA)
### SPAM, Work, and Personal Phrase Lists

- **General Spam (Bit 0)**
  - “amazing”
  - “CALL NOW”
  - “Limited Time Offer”
- **Save Money SPAM (Bit 1)**
  - “Consolidate”
  - “full refund”
- **Fast Money SPAM (Bit 2)**
  - “MAKE MONEY FAST”
  - “Work from home”
- **Chains and Forwards (Bit 3)**
  - “Read this”
  - “FWD”
- **Jokes (Bit 4)**
  - “Joke”
  - “walks into bar”
- **Work List (Bit 5)**
  - “Homework”
  - “Machine problem”
  - “CS536”
  - “Lockwood”
  - “Washington University”
- **Personal List (Bit 6)**
  - “Mom”
  - “Dad”
  - “Call Home”
- **Urgent (Bit 7)**
  - “Urgent”
  - “Emergency”

**Note:** Underscored letters are case-insensitive

### Content Matching Module

```
regex_app (given)
```

- `dataen_out_appl` → `dataen_appl_in`
- `d_out_appl` → `d_appl_in`
- `sof_out_appl` → `sof_appl_in`
- `eof_out_appl` → `eof_appl_in`
- `sod_out_appl` → `sod_appl_in`
- `tca_out_appl` → `tca_appl_in`

Inputs:
- `clk`
- `reset_l`
- `enable_l`

Outputs:
- `Matcher`
- `ready_l`

From Protocol Wrappers

To existing MP1 circuit

To extended Bits of CAM

`wrapper_module.vhd`
### Content Match Vector

- Bit is set (1) if any phrase in a category is found anywhere in payload
- Bit is clear (0) if none of the phrases in the category appear in the payload

### Sample Content Match Vector

- “Consolidate your loans. CALL NOW”
Packet matching w/ Content Addressable Memory

- Sample Packet:
  - Source Address = 128.252.5.5 *(dotted.decimal)*
  - Destination Address = 141.142.2.2 *(dotted.decimal)*
  - Source Port = 4096 *(decimal)*
  - Destination Port = 80 *(decimal)*
  - Protocol = TCP (6)
  - Payload = “Consolidate your loans. CALL NOW”
    - Payload Lists = { General SPAM (0), Save Money SPAM (1) }
    - Content Vector = “00000011” *(binary)* = x”03” *(hex)*

<table>
<thead>
<tr>
<th>Content</th>
<th>Src IP (hex) =</th>
<th>Dest IP (hex) =</th>
<th>Src Port =</th>
<th>Dest Port =</th>
<th>Proto =</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>80FC0505</td>
<td>8DBE0202</td>
<td>1000</td>
<td>0050</td>
<td>06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Mask:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 = care</td>
<td>0 = don’t care</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>DROP the packet: It matches the filter</td>
</tr>
</tbody>
</table>
### Basic Packet Classification Hardware

- CAM MASK 1
  - CAM VALUE 1
    - CAM VALUE 2
      - CAM MASK 2

![Diagram of packet classification hardware](image)

match <= '1' when (input = 0xFFFFFFFFFFFFFFFFFFFFFFF)
else '0';

### Quality of Service

- **Gives some flows better service than others**
  - Zero-sum game:
    - The amount of bandwidth on a link is fixed.
  - Design the system to maximize reward
    - Not all Internet users are created equal.
    - Will users pay for something that is not well defined?
  - Some flows get worse service!

- **Applications**
  - Determine which flows to drop under heavy load
  - Try to provide virtual link with end-to-end guarantee on bandwidth and delay
Data Queuing Outline

- **Types of Queuing**
  - Class-based
    - Priority-based Service Disciplines
    - Differentiated Services
  - Per-Flow
    - Fluid Fair Queuing / Weighted Fair Queuing
    - Constant-Time Queue Service Disciplines

- **Hardware Implementation of Flow Queues**
  - Running time limitations in Gigabit networks

Queue Management

- **Flow State**
  - Head Pointer
  - Packet Reads = 0
  - Packet Writes = 3
  - Tail Pointer

- **F_i**
  - Other Flow State
  - Head Pointer

- **M[x]**
  - Reserved Empty Slot [implementation dependant]

- **M[y]**
  - Head Pointer

- **M[z]**
  - Tail Pointer

- **M[u]**
  - Empty Slot

- **M[v]**

Note that packets can be stored anywhere in memory.
The 3DQ Scheduler for MP3 combines priority-based scheduling with flow-based scheduling

- Supports four priority levels
  - Each implemented as a CoreGen FIFO that stores Flow IDs

- Supports per-flow queuing
  - Backlogged flows are serviced in a round-robin manner
**Extensible Interface to Flow Buffer**

--- Interface with the Queue Context

--- From the Queue Manager
Tail: in std_logic_vector(31 downto 0);
TailValid: in std_logic;

--- To the Queue Manager
NextTail: out std_logic_vector(31 downto 0);
NextTailValid: out std_logic;

--- From the Queue Manager
Head: in std_logic_vector(31 downto 0);
HeadValid: in std_logic;

--- To the Queue Manager
NextRead: out std_logic_vector(31 downto 0);
NextReadValid: out std_logic;

**Content-Aware Firewall on a Chip**

(Implemented on the RAD on the FPX, using a VirtexE 2000 FPGA)
**DHP Design Flow**

- **Synthesize Logic** (Synplicity Pro)
- **Constrain Placement to modular Regions (DHP)**
- **Functional Verification (ModelSim)**
- **Place and Route with constraints (Intel)**
- **Set Boundary I/O & Routing Constraints (DHP)**
- **In-System, At-Speed Testing (FPX Platform)**
- **Install Module and Generate Traffic (NCHARGE)**
- **Relocate Modular Logic within bitstream (PARBIT)**
- **Verify Post Place & Route Timing (ModelSim)**
- **Generate bitstream (Intel)**

**Packet Processing Waveform**

- TTL
- Src IP
- Dest IP

506 ns to 706 ns
Synthesizing the Design (Synplicity)

Listing of the synthesizable vhdl files in the design. The files are listed in bottom-up order.

Target FPGA
Xilinx Virtex-E XCV2000E -7 FG680

DHP Implementation on the FPX

- Programming Module onto the Reprogrammable Application Device (RAD)
  - Circuit on XCV2000E
  - SRAM and SDRAM Interfaces
  - Control Cell Processor
  - Holds 2+ DHP Modules
  - Reprogrammable over network
Floorplannings - INFRA

DHP RESERVED AREAS

NCHARGE Homepage

- Web interface to communicate with the FPX

- Menu Functions
  - Route traffic flows
  - Reprogram hardware
  - Upload bitfiles
  - Read & write on-chip Memory
  - Create test cells
GUI to Install New Hardware Module

- Allows uploading and downloading of full or partial bitfiles
- Allows user to select a bitfile for programming
- FPGA reconfigures when user presses ‘Execute Command’

Hardware Plugin Service Model

- Traffic Source
- Service Interface
- Data Flow 1 (with QoS specifications)
- MultiService Router, R1
- Service Interfaces
- Plugin
- Reconfigurable hardware module
- Software module
- Plugin Server
- Backpressure/flow control
- Data Flow 2 (with QoS specifications)
- MultiService Router, R2
- Service Interfaces
- Plugin
- Reconfigurable hardware or software module
- Backpressure/Flow control
- Data Flow 3 (with QoS specifications)
- Service Interface
- Destination
**Example: Secure, Reliable, Video Distribution**

![Diagram of secure, reliable video distribution]

---

**Reprogramming the FPX**

1. **New Module is created**
2. The full or partial bitstream is sent over network to the NID on the FPX and stored in the configuration cache
3. A Command Issued to reconfigure hardware
4. NID Reads Memory and reprograms RAD via SelectMAP

---

Washington University in St. Louis
Simultaneous Programming Within a Switch

Interfacing with Computers & Internet
FPX Platform platform developed

FPX combined with WUGS to route and process Internet packets

Other Modules Implemented

- IPv6 Tunneling Module
  - Tunnels IPv6 over IPv4

- Statistics Module
  - Event counter

- Traffic Generator
  - Per-flow mixing

- Video Recorder
  - Motion JPEG

- Embedded Processor
  - KCPSM

- IPv4 CAM Filter
  - 104 Bit header matching

- Fast IP Lookup (FIPL)
  - Longest Prefix Match
  - MAE-West at 10M pkts/second

- Packet Content Scanner
  - Reg. Expression Search

- Data Queueing
  - Per-flow queue in SDRAM
Summary

- **Extensible Network Technology**
  - Provides high degree of configurability
  - Provides high performance

- **Field Programmable Port Extender (FPX) Platform**
  - Well-defined Application Programming Interface
  - Web interface simplifies FPX control and configuration
  - 85 FPX devices built
  - Over 30 Hardware applications developed & tested on FPX

- **Applications**
  - Internet Protocol (IP) Protocol Processing
  - Network Security (Firewall)
  - Quality of Service (Flow Scheduler)
  - Multi-video processing Platform
  - Your modules!

FPX used for Hands-on Tutorials

Gigabit Kits workshops with FPX Tutorials held 1/01, 8/01, 1/02, & 6/02

Participants Programmed FPX Modules in the CAD Laboratory

Then tested modules in lab

Acknowledgements

- Washington University
  - Applied Research Lab
    - Faculty
      - John Lockwood
      - Jon Turner
    - Graduate Students
      - David Taylor
      - Todd Sproull
      - Sarang Dharmapurikar
      - David Lim
      - James Moscola
      - David Schuehler
      - Chris Neely
      - Chris Zuver
      - Haoyu Song
      - Henry Fu (Now at Stanford)
      - Bharath Madhusudan
    - Undergraduate Students
      - Harvey Ku (at CMU)
      - Elliot Sinclair
      - Mike Attig
      - Doug Stirrut
      - Tucker Evans (Now at General Dynamics)
      - Mike Wrighton (Now at CalTech)
    - Industry Research Partners
      - David Parlour (Xilinx)
      - Matthew Kulig (Global Velocity)
- University Research Partners
  - Prabhu Kuttyiam (University of Kentucky)
  - Ken Calvert (University of Kentucky)
  - Matt Sanders (Georgia Tech)
  - Ron Srodawa (Oakland University)
  - Haiyan Qiao (NDSU)
  - William Perrizo (NDSU)
  - Kuo-Tung Kuo (University of Maryland)
  - Cary Colwell (Naval Postgraduate School)
  - John Gibson (Naval Postgraduate School)
  - Hualyu Liu (University of Texas at Austin)
  - Qing Tan (University of Toledo)
  - Sachin Shetty (University of Toledo)
  - Rajanikanth Batchu (Mississippi State)
  - Ravi Sankar (USF)
  - Simon Wong (UCLA)
  - Sven Shepstone (University of Cape Town, South Africa)
- Visiting Faculty and Students
  - Edson Horta (Univ. de Sao Paulo, Brasil)
  - Florian Braun (University of Stuttgart)
  - Carlos Macian (University of Stuttgart)