MPhil in Advanced Computer Science
Building an Internet Router (revised for 2010)

Leader: Andrew W. Moore
Timing: Michaelmas
Structure: 1 Introductory lecture, 1 × 1-hour orientation
15 × 2-hour supervised laboratory slots

Prerequisites: Digital Communications I & II; C/C++, ECAD/ECAD-Practical, Unix Tools

However, given we also take students that are not from the CST system, the prerequisite requirements are spelled out more fully:

- All students will require an understanding of IP-level networking. Equivalent course requirement is Digital Communications I and II.
- All students are expected to be competent UNIX users; this includes familiarity with make and the perl programming language. The equivalent course requirement is *Unix Tools*.
- and at least one of:
  - Students who are familiar with Verilog (or VHDL) and are comfortable with the general process of designing RTL-based logic and the verification process associated with that. The student must understand the difference between behavioral Verilog and RTL Verilog (a subset which can be synthesized into logic). Equivalent hardware course requirement is *ECAD/ECAD-Practical*.

    OR
  - Students who are competent software programmers, who have had significant exposure to system level programming in the C programming language (ideally under UNIX). The students must be comfortable developing and debugging in a multithreaded environment, and should have experience working within a code-base of over 5,000 lines. Equivalent software course requirement is *Programming in C/C++* along with a significant project in C/C++

AIMS

This module requires the full design, implementation, testing and inter-operability of a complex hardware and software system.

SYLLABUS

This subject is lab-centric and project-based. The first class is the only scheduled lecture. During the term, groups of two or three students will work together to

develop a fully functional IP router. The groups will consist of at least one student familiar with designing hardware in Verilog and one student who is comfortable writing large, system-level network programs in C. Students will be paired by area on the first day of class.
The hardware uses the NetFPGA boards which provide a programmable hardware platform for developing network equipment. Given the Verilog HDL code for a simple four port switch the hardware designer will extend/modify/discard this code to provide the functionality of a four-port IP router. A set of tools are provided to assist the student with design, verification and synthesis.
Each group will:

- design and implement a router in 8 weeks,
- an inter-router protocol (PW-OSPF),
- show router interoperability with other groups, and
- extend the basic router with new features (e.g., Firewall, NAT, Fairly-resourced Output queues, Packet capture, Traffic generator).

OBJECTIVES

By the end of this course, all students will be able to:

- Describe what responsibilities a user-level application, operating system, or device driver has in communication. (For example, a web browser talking to a web server)
- Identify the challenges inherent in efficiently transferring data from the general-purpose machine to the network peripheral.
- Compare and contrast a network interface built with an embedded processor versus dedicated (fixed) hardware on the basis of design time (up-front expenses), manufacturing cost (per-unit), performance, functionality, etc...
- Explain how the OSPF routing protocol enables communication when presented with a new network topology
- Describe how the spanning tree Ethernet protocol prevents circular paths from developing in computer networks.

By the end of this course, students choosing to focus on network hardware will be able to:

- Create an FPGA-based hardware router that implements longest-prefix-matching lookups
- Explain the required functions that the hardware router must provide to enable basic network functionality

- Determine which functionality should be implemented in hardware versus software
- Evaluate different implementations that can achieve the basic functions
- Optimize their FPGA design for both logic (device utilization) and performance (network throughput)
- Integrate the hardware router with control software being co-developed by other group members

By the end of this course, students choosing to focus on network software will be able to:

- Create a program that controls and manages the hardware router being implemented by other group members
- Explain the required functions that the router must provide to enable basic network functionality
- Determine which functionality should be implemented in software versus hardware
- Design a control program using the Virtual Network System that participates in the PW-OSPF dynamic routing protocol and responds to ARP and ICMP messages.
- Design a control program using the Virtual Network System that exports a command-line interface and allows a user to inspect and modify the state of the hardware router.
- Integrate the control program with the hardware router being co-developed by other group members

COURSEWORK

Not Applicable - course consists entirely of the assessed practical work.

PRACTICAL WORK

Two supervised laboratory slots are scheduled per week however, additional laboratory work will be required to completed this project. In addition to ad-hoc availability, each group will have one-on-one time with the lecturer for a fixed slot each week to track progress.
Additional time is provided both for small-group tutorials on specialist skills, not present in all the incumbent students. Precise material to be covered is heavily dictated by the students on the course but material has included: perl, unix, make, advanced verilog, digital design including metastability issues, software timer loops, protocol implementation, dikjstra's algorithm, along with using debugging tools such as gdb and modelsim.

Deliverables are due at the end of each week, exact details of dates and mechanism will be provided at the time of the module.

— **[Week 1]** —

Software: Develop a functional router in software

SW Deliverable: Design document with overview of router architecture

Hardware: Verify tools, review provided code and develop hardware architecture for your complete router. Implementation of four-port non-learning switch

HW Deliverable: Email summarizing your progress getting tools set up:Simulation, Synthesis and running on the hardware. Working four-port non-learning switch (bit-file and archive of project directory)

— **[Week 2]** —

Software: Develop a functional router in software (continued), Adding a command line interface to your router

SW Deliverable: Working software router, Add CLI design and specifications to the design document, Software router with integrated CLI

Hardware: Develop a four-port learning Ethernet switch

HW Deliverable: Working four-port learning switch (bit-file) First version of Hardware Design Document, Test Summary

— **[Week 3]** —

Software: Develop router-router protocol in software router

SW Deliverable: Router-router protocol section of the design document

Hardware: Complete four-port learning Ethernet switch with packet identification

HW Deliverable: Verification section of the Hardware Design Document, bit-file for a working four-port learning switch with the ability to count different types of packets

Integration: determine what extra registers (if any) you are going to support and how to handle them from within the CLI

— **[Week 4]** —

Software: Develop router-router protocol in software router (continued)

SW Deliverable: Software Router with fully implemented router-router protocol

Hardware: Implement forwarding path to-from software

HW Deliverable: Bit-file of design, should forward all appropriate packets to the software and forward packets from software to the correct ports

Integration: verify that the hardware and software are able to correctly exchange packets

— **[Week 5]** —

Software: Add Hardware Control to Router

SW Deliverable: Add support for register access to design document, Software router with register access support

Hardware: Working IP router

HW Deliverable: bit-file for the working router, updated Hardware Design Document (with router interoperability testing section)

Integration: use software to interrogate/test operation of hardware

— **[Week 6]** —

Software: Advanced feature development and router interoperation

SW Deliverable: Design document with full specifications of proposed advanced functionality
Hardware: Router interoperation and advanced feature development
HW Deliverable: bit-file and tarball of the functional router
Integration: Integrate router and hardware for fully functional router, Inter-operate with 2 or more other groups.
— [**Week 7**] —
Software: Develop advanced functionality
SW Deliverable: Proof of progress on advanced functionality
Hardware: Continue implementing advanced functionality, Enhance algorithms
HW Deliverable: Email update of progress, Updated hardware design document (including advanced feature section)
— [**Week 8**] —
Software: Finish advanced functionality
SW Deliverable: Submissions of router with advanced functionality implementation, Completed design document for software portion.
Hardware: Finish enhancing algorithms/interoperation and implementing advanced features
HW Deliverable: bin file and source files, updated hardware design document with your added functionality.
Group Deliverables:
Activity report: A single copy of a printed report signed by all group members that describes what each group member contributed to the project (code, documentation, testing, co-ordination....) *There are no points for this document, but points will be deducted if it is not provided.*
Presentation: A demonstration of the advanced features of the group and a presentation by each group covering the architecture of their design (hw and sw), the problems encountered, etc. Approximately 20 minutes per group.

ASSESSMENT

This subject is marked 0–100; 60 is the pass mark.

Marks	Item
10	Participation. This will be a subjective judgment by us based on aspects such as: our interaction with you in regular meetings and your team's final write-up on who did what.
10	Interoperability. 5 points if your router interoperates in a topology of our choosing with several instantiations of our reference router. 2 additional point for each other team's router you interoperate with (under the same conditions), to a maximum of two teams (i.e. 4 points max).
65	Functionality: deliverables are described in the Schedule table above
15	Documentation: deliverables are described in the Schedule table above
Up to 10 bonus marks	Presentation: Marks for clarity, technical material and for providing insights into the issues you encountered and how you overcame them (or didn't).

The presentations will be made in front of a group of panelists drawn primarily from industry; their input will be sought on the quality of the presentation. The presentation will not cause marks to decrease — only increase.

The Carrot

The judging panelists will make a non-trivial judges award to the best group. This award is intended as additional incentive and will not directly impact the assessed outcome.

The Stick

Standard rules of mark-decimation will apply for late deliverables; refer to the MPhil ACS web page on assessment for details.

RECOMMENDED READING

* Strazisar, V. (1979), How to build a gateway, `http://www.networksorcery.com/enp/ien/ien109.txt`

Postel, J. (ed) (1981), Internet Protocol, `http://www.ietf.org/rfc/rfc0791.txt`

Baker, F. (ed) (1995), Requirements for IP Version 4 Routers, `http://www.apps.ietf.org/rfc/rfc1812.html`

Comer D. (2006) Internetworking with TCP-IP, Vol 1.

OTHER :

Comer, D. & Stevens, D. (1995). Internetworking with TCP-IP, vol. 2. Prentice Hall (3rd ed.).

Peterson, L.L. & Davie, B.S. (2007). Computer networks: a systems approach. Morgan Kaufmann (4th ed.).

Varghese, G (2005). Network Algorithmics. Morgan Kaufmann (1st Ed.).

Last updated: March 2010