

# MPhil in Advanced Computer Science

## Building an Internet Router

**Leader:** Andrew W. Moore

**Timing:** Michaelmas

**Structure:** 1 Introductory lecture, 8 × 2-hour supervised laboratory slots

**Prerequisites:** Digital Communications I & II; C/C++, ECAD/ECAD-Practical  
To assist those applicants from outside Cambridge, the prerequisite requirements are spelled out more fully:

- Students will require an understanding of IP-level networking. Equivalent course requirement is Digital Communications I and II.
- 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.
- Students who are competent software programmers, who have had significant exposure to system level programming in C. 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 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,

- write software for Command Line Interface (CLI) and 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
- Integrate the hardware router with control software being co-developed by other group member(s)

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 member(s)

- 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 member(s)

## COURSEWORK

Not Applicable; this course consists entirely of assessed practical work.

## PRACTICAL WORK

One supervised laboratory slot is scheduled per week however, additional laboratory work will be required to complete this project. Additionally, each group will have one-on-one time with the lecturer for a fixed slot each week to track progress. Deliverables are due at the end of each Week, exact details of the submission mechanism will be provided at the time of the module.

### Deliverables Schedule

Week	Software	Hardware	Deliver	Due date
1	Verify Software Tools	Verify CAD Tools	Design Document	15/Oct/09
2	Build Software Router	Build Non-Learning Switch	Runnable Software Router	22/Oct/09
3	Command Line Interface	Building Learning Switch	Runnable Basic Switch	29/Oct/09
4	Develop Inter-Router Protocols	Output Queues	Runnable Learning Switch	5/Nov/09
5	Implement Inter-Router Protocols	Forwarding Paths	Interface S/W & H/W	12/Nov/09
6	Interoperate Software & Hardware		Router Submission	19/Nov/09
7	Plan New Advanced Feature		Design Plan	26/Nov/09
8	Demonstrate New Advanced Feature		Demonstration	3/Dec/09

**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 presentation by each group covering the architecture of their design (hardware and software), 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 panelists will make a non-trivial 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

### Preparatory Reading

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

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

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

### Further Reading

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

Comer, D. & Stevens, D. Internetworking with TCP-IP, vol. 2. Prentice Hall

Varghese, G Network Algorithmics. Morgan Kaufmann

Last updated: January 2009