Diploma Theses for the 2009-2010 period

Korkakakis Nikos* – Vlachos Kyriakos†

July 17, 2009

Abstract

In this document are proposed some interesting theses closely/loosely coupled with optical testbed work in progress.

1 Proposing IPv4/v6 compatible routing schemes within a GMPLS environment

In this work the student(s) should propose some new dynamic routing schemes that take advantages of the GMPLS infrastructure developed in the photonics lab. The idea is to design and implement new routing policies, that may coexist with an optical (or other) network managed by GMPLS. The student(s) are required to propose new ideas about routing in ultra highspeed network capable of quick rearrangments in case of fault occurence. The working environment is any standard Linux with up to date kernel source and their choice of one of the following userspace routing daemons;

1. QUAGGA – written in C – Quagga website
2. XORP – written in C++ – XORP website
3. BIRD – written in C – Bird website

Student(s) should extend at some point one of the daemons capabilities adding the ideas they propose in their thesis. It is obvious that the student should have some knowledge for networking (IP based packet switched netorks), some programming skills in the respective daemon's language, and the will to get hands dirty.

2 Extending ns3 to real world networks

One of the greatest features the ns3 has is th realtime packet scheduler. This means that functonality such as simulator based packet generation with real hardware is true as shown in SIGCOMM 2008. The task in this thesis is to replicate the work done in SIGCOMM concerning ns, for interoperation with the actual optical testbed (create analogous software hooks that may let the ns3 interoperate with the testbed), and to evaluate the performance of the simulator and its realtime capabilities. The major questions that will be examined are;

*korkakak@ceid.upatras.gr
†kvlachos@ceid.upatras.gr

- NS3 real time scheduler
- Driving real hw with NS3
- NS3 cradle intergration

Figure 1: List of interesting links

1. Can ns3 "drive" a full 10GBps network link?

   If not what hardware would be suitable to do so?

2. Is there a theoretical bound to ns3 performance running on the real time scheduler?

3. Can we use extra ns3 functionality to feed the actual testbed with real world application data streams?

   If yes can we develop it?

In this work the student should be able to cope with the development of ns3 code so familiarity with networking, and programming languages (C++) is needed.

3 Implementation of RSVP-TE

In this thesis the student(s) has to implement the RSVP-TE[1] protocol in one of the following daemons

1. QUAGGA – written in C – Quagga website
2. XORP – written in C++ – XORP website
3. BIRD – written in C – Bird website

Student(s) should extend at some point one of the daemons capabilities adding the RSVP-TE protocol. It is obvious that the student should have some knowledge for networking (IP based packet switched netorks), some programming skills in the respective daemon's language, and the will to get hands dirty.

[1] See also RFC 3209, RFC 4090, RFC 4920, RFC 5151