The ESLEA Control Plane Software

A.C. Davenhall,[†] P. Clarke[†] & N. Pezzi[‡]

[†] National e-Science Centre, Edinburgh
[‡] Department of Physics and Astronomy, University College London

Abstract

Circuit-switched networks are emerging as an attractive alternative to conventional packet-switched ones for some types of application, particularly ones involving transferring a large volume of data. The ESLEA Project will use the UKLight circuit-switched network, and its analogues overseas, to demonstrate the usefulness of circuit-switched networks in several applications disciplines. A circuit-switched network must be configured to establish a circuit before it can be used, and the Control Plane Software (CPS) is ESLEA's software for this purpose. The requirements, design and implementation of the CPS are described. One notable feature of the CPS is that it will allow users to specify reservations for circuits in advance. It will then automatically create the required circuit at the specified time. The current status of the CPS is reviewed and future plans are discussed.

1 Introduction

Circuit-switched networks are emerging as an attractive alternative to conventional packet-switched networks for some types of application. They are suitable for applications which need to transfer a large volume of data or which have some other demanding requirement, such as low latency or a constant transfer rate (that is, low jitter) and which operate between a fixed set of end-points.

In recent years an international effort has developed to investigate and use such circuit-switched networks. This initiative began with the NetherLight[1], StarLight[2] and CANARIE[3] Projects, respectively in the Netherlands, the US and Canada, but now includes many other contributors. These various national projects collaborate through the Global Lambda Infrastructure facility (GLIF).[4] The UKLight network[5] is the UK contribution to this emerging infrastructure.

The ESLEA (Exploitation of Switched Light Paths for e-Science Applications)[1][6] Project will use UKLight, and its analogues overseas, to demonstrate the usefulness of switched-circuit networks in real applications and to gain practical experience of their use. This exercise is timely because many of the next generation of production academic networks, including SuperJanet5[7] and GÉANT-2[8], will include some switched-circuits amongst their facilities.

In a circuit-switched network a dedicated connection (or 'circuit') is established from a source to a destination host. Before data can be transferred the circuit must be set up, and similarly it should be dismantled once the transfer is complete (that is, it should be respectively 'nailed-up' and 'torn down' in the argot of computer networks). Software is required to perform these operations and in the case of ESLEA this software is the Control Plane Software (hereinafter CPS). This paper first gives a brief overview of ESLEA in sufficient detail to understand the CPS. It then outlines the requirements and design of the CPS and discusses the current state of its implementation and the future plans.

[1] See URL: `http://www.surfnet.nl/innovatie/netherlight/`
[2] See URL: `http://www.startap.net/starlight/`
[3] See URL: `http://www.canarie.ca/`
[4] See URL: `http://www.glif.is`
[5] See URL: `http://www.uklight.ac.uk/`
[6] See URL: `http://www.eslea.uklight.ac.uk/`
[7] See URL: `http://www.ja.net/SJ5/`
[8] See URL: `http://www.geant2.net/`

2 The ESLEA Project

The ESLEA project started in February 2005 and will run for two years. It will demonstrate the usefulness of circuit-switched networks by running so-called 'exploitation' sub-projects in four applications disciplines (see Table 1). To this end it involves collaborations between various groups working in these diverse fields (see Table 2).

Exploitation projects	Discipline
HEP	High-energy physics
e-VLBI	Radio astronomy
ReG	RealityGrid; interactive simulation and modelling
e-Health	Medical applications

Table 1: ESLEA exploitation projects. See Greenwood *et al.*[1] for further details

In addition to the four exploitation projects ESLEA also has two so-called 'capability development' projects which will develop the infrastructure necessary for the exploitation projects. One capability development project will develop high-performance versions of the TCP protocol optimised for use with circuit-switched networks. This work is not mentioned further here. The second will develop the CPS software which is the subject of this paper.

2.1 The ESLEA network

The distribution of the various groups participating in ESLEA is such that elements of the four exploitation projects are present in several universities and laboratories (see Table 3). ESLEA has located a Cisco 7609 switch/router at most of these institutes. These machines will be used to connect the local end-hosts to the UKLight access point (typically the end-hosts and the 7609 would be on a local LAN). The 7609s were inherited from MB-NG (Managed Bandwidth - Next Generation)[9], an earlier project which is now complete. In a couple of cases it has been necessary to relocate them. The 7609s will be configured to switch traffic from the specified end-hosts into the UKLight connections. This arrangement provides the necessary capability to construct circuits linking end-hosts across UKLight.

[9]See URL: `http://www.eslea.uklight.ac.uk/mb-ng/`

CCLRC
Daresbury Laboratory
Rutherford Appleton Laboratory

National e-Science Centre

University College London
Department of Chemistry
Department of Computer Science
Department of Physics and Astronomy

University of Lancaster
Department of Computing
Department of Physics

University of Manchester
Manchester Computing
School of Physics and Astronomy

University of Oxford
Computing Laboratory

Table 2: Departments and groups participating in ESLEA (in alphabetical order)

Site	Exploitation projects	Cisco 7609s
UCL	HEP ReG	2
Manchester	HEP ReG e-VLBI e-Health	2
RAL	HEP ReG e-Health	1
Lancaster	HEP ReG e-Health	1

Table 3: Exploitation projects present in the various ESLEA institutions and the location of the Cisco 7609 switch/routers. Note that there are no 7609s located at NeSC, Oxford or the Daresbury Laboratory

The requirements of the various exploitation projects vary. We envisage that some of the circuits will be permanent. Others will be created and torn down on demand. The detailed requirements of individual exploitation projects are not germane here. However, the purpose of the CPS is to allow the users to create temporary circuits when they desire.

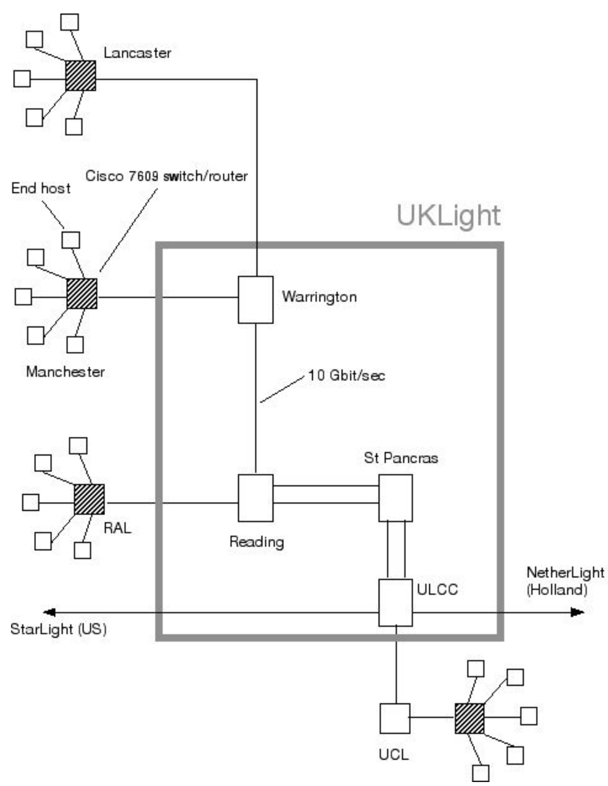

Figure 1: Topology of the basic ESLEA network. Only 'edge-switching' is available in the Cisco 7609s around the periphery of UKLight

Figure 1 shows a simplified schematic version of the basic ESLEA network, as described above. All connections are provided by configuring the 7609s. That is, switching is provided purely around the periphery of UKLight. Switching within the core of UKLight is not available. This arrangement might be adequate. However, additional flexibility will probably be required, which we will provide by simulating switching in the UKLight core. All the connections from the various 7609s would be routed to the University of London Computer Centre (ULCC) and thence to a switch outside UKLight (see Figure 2). That is, this switch, though physically outside UKLight would be logically within it. It is not yet entirely clear whether this additional flexibility will be required, but it seems likely. If it is needed we would prefer to use an optical rather than an electronic switch and we hope to collaborate with the Photonics Networks Laboratory[10] at the University of Essex in its provision.

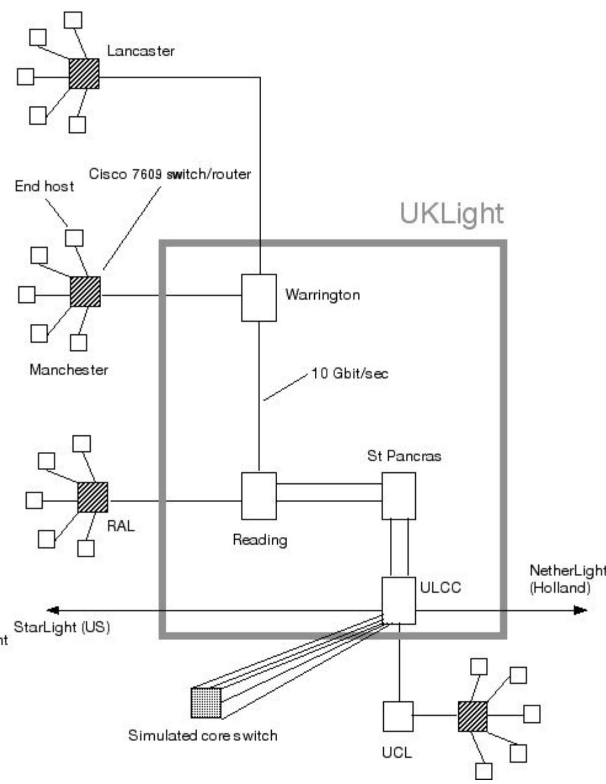

Figure 2: Topology of the ESLEA network with simulated switching in the UKLight core. All links from the Cisco 7609s pass through ULCC to the simulated core switch, which is physically outside UKlight

3 Requirements

The purpose of the CPS is to allow users to establish circuits between end-hosts. In the simple case of the basic ESLEA network all that is required to establish a circuit is to configure, as appropriate, one or both of the 7609s connecting the end-hosts to UKlight. If the simulated central switch is also available then it too must be controlled.

The CPS software will be used, at least initially, within the ESLEA Project. Consequently, it will

[10] See URL: `http://www.essex.ac.uk/ese/research/pn_lab/`

have only a limited number of users. Further, we anticipate that it will mostly be used by network administrators, or people in similar positions, rather than genuine scientific end-users. Despite having this limited and rather expert user-base, it is nonetheless desirable for the software to be flexible, simple to use and secure. The users are geographically dispersed (as are the network switches, of course).

Basic software to create circuits in real time is required. However, we intend that circuits will usually be 'reserved' in advance. That is, the user specifies the circuit before it is needed and the CPS checks whether it can be created (sufficient links are available between the given end-hosts *etc*). If the circuit can be created the CPS stores the details of the reservation and subsequently creates the circuit at the time requested by the user. When checking whether a given reservation can be accepted the CPS should take cognizance of the requirements of any previous reservations in order to avoid conflicts when the circuits are created. The user should be able to examine, modify and delete pending reservations.

The user will supply a 'high-level' description of a reservation, comprising, for example, the start and destination hosts, the start and end times and the capacity required, rather than giving a detailed description of individual circuits. The basic information to specify a reservation is listed in Table 4. Even more abstract specifications of a reservation might be possible. For example, in some applications, such as file transfer, it might be adequate to specify merely the end-hosts, the start and end times and the total volume of data to be transferred. The CPS is then free to arrange sufficient circuits to transfer these data as it chooses, constrained only to transfer the data between the given times.

It will be possible to invoke the CPS from a GUI, by submitting text files or scripts containing a description of a reservation (to permit 'batch' type operations) and from a programmatic interface. The latter option will allow programs to make reservations automatically. Figure 3 shows one possible appearance for the GUI and Figure 4 a form for the text descriptions.

Details of past reservations should be retained by the system and used to derive statistics about the usage of the ESLEA network.

Authentication, authorisation and accounting (AAA) is required to ensure that users are known to the system and authorised to perform the operations that they are attempting. Users should be able to control who is able to examine, modify and

Source host
Destination host
Start time and date
Stop time and date
Throughput
Service type
Reverse fraction

Table 4: Basic details to specify a reservation. 'Service type' is an indication of the type of service required, such as a bulk (or file) transfer or a continuous data rate. 'Reverse fraction' is the capacity required for acknowledgements *etc*.

```
#+
# Example reservation definition file.
#
# A.C. Davenhall (Edinburgh), 28/4/05.
#-

#
# A reservation where all the values
# have been supplied.

reservation
  Owner            ACD
  Source           host2.cs.man.ac.uk
  Destination      host3.hep.ucl.ac.uk
  StartTime        2005-05-01-18:30
  StopTime         2005-05-01-19:30
  Throughput       1
  ServiceType      bulk
  ReverseFraction  10
endreservation

#
# A reservation where some of the values
# have been defaulted.

reservation
  Owner            ACD
  Source           host2.cs.man.ac.uk
  Destination      host3.hep.ucl.ac.uk
  StartTime        2005-05-02-18:30
  StopTime         2005-05-02-19:30
endreservation
```

Figure 4: A possible form for a text-file specification of a reservation

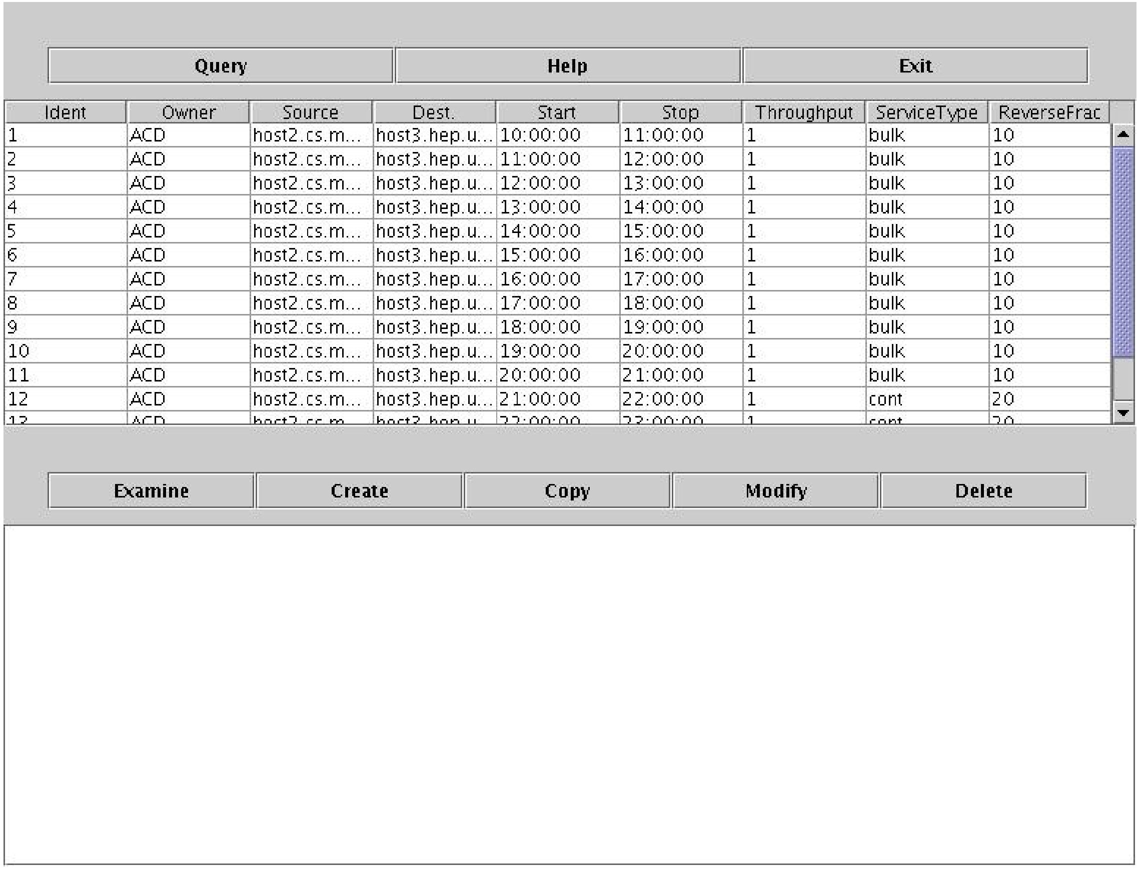

Figure 3: One possible appearance for the CPS GUI. Pending reservations are displayed in the upper part of the window, which may show either all pending reservations or a subset of them selected according to some criterion. Clicking on a reservation displays additional information about it. The surrounding buttons allow reservations to be created, modified and deleted or a new set of reservations to be displayed. The bottom part of the window is for informational and error messages.

delete their reservations. The default permissions might be that all users can see a reservation, but only the owner can modify or delete it. However, the permissions should be under the control of the owner: there might be cases, for example, where a user wishes to prevent other users from examining his reservations or where a group of collaborating users wish to be able to modify and delete each others' reservations. A system administrator should be able to over-ride a user's permissions. These features are, of course, simple analogues to the control over access to files provided by operating systems. The CPS should also keep track of each user's usage and deduct it from a personal quota, in order to prevent one user cornering the system to the detriment of others.

4 Implementation Strategy

The CPS is implemented as a modification of the Network Resource Scheduling (NRS)[11] software developed by Bhatti *et al.* at UCL (see Rio *et al.*[2] for a description of NRS). NRS provides facilities for reserving guaranteed throughput over conventional packet-switched networks. It uses the DiffServ (Differential Service) mechanism to deliver the guaranteed throughput. Bandwidth allocation reservation (BAR) systems, such as NRS, have much in common with the CPS: both deliver a guaranteed service to the user, though the mechanisms by which they provide it are very different. NRS has most of the features required for the CPS, particularly the advance specification of reservations.

[11] See URL: `http://www.cs.ucl.ac.uk/staff/S.Bhatti/grs/`

The basic architecture of NRS is shown in Figure 5. NRS is a distributed system with no central control, an arrangement which is both robust and extensible. There is one NRS instance per Cisco 7609 and this NRS superintends all the connections made through the 7609 and maintains a database where the reservations are stored. The procedure to make a new reservation is as follows.

1. The user invokes the NRS client and specifies the reservation required.

2. The client sends the reservation to the local instance of NRS.

3. The local NRS checks whether it can satisfy the reservation. It determines whether it has adequate capacity by checking its reservation database for other reservations which are discharging at the same time.

4. It also invokes the remote NRS (the one superintending the 7609 to which the remote end-host is connected). This NRS checks whether it too can satisfy the reservation.

5. If both NRS instances can satisfy the reservation it is accepted and entered into both their databases. If either cannot satisfy it it is rejected. The user is notified appropriately.

6. When the start time arrives both NRS instances configure their 7609s to construct the circuit.

The GUI/client (see Figure 5) is a separate program which can be remote from the NRS. NRS is written in Java and stores the list of reservations using the database PostgreSQL.[12] The components communicate using the Beep protocol.[13] NRS also uses a few other open-source packages.

Ultimately the Cisco 7609s are configured using commands in Cisco's IOS operating system. The switch controller 'backend' of NRS (see Figure 5) assembles the necessary commands and sends them to the 7609. In practice only a few IOS commands are needed to configure the 7609s. The bulk of NRS is 'high-level' Java code concerned with manipulating reservations.

If a simulated core switch is used NRS will need to be modified to control it. Here again we plan to adapt work on BAR systems. The model is that systems around the periphery of a network which wish to make reservations communicate requests to the network through an interface. Software internal to the network processes the request, determines whether or not the reservation can be accepted and replies accordingly. There are two advantages to this approach.

1. The network behind the interface can be of arbitrary complexity (though in our case it would be a single switch) and this complexity is hidden.

2. Usually the network is outside the control of the hosts around the periphery, so a model in which they make requests of it (rather than sending commands to configure it) is necessary.

Suitable interfaces have been specified by Patil *et al.*[3] as part of the BAR work undertaken in the Enabling Grids for E-Science (EGEE) Project.[14]

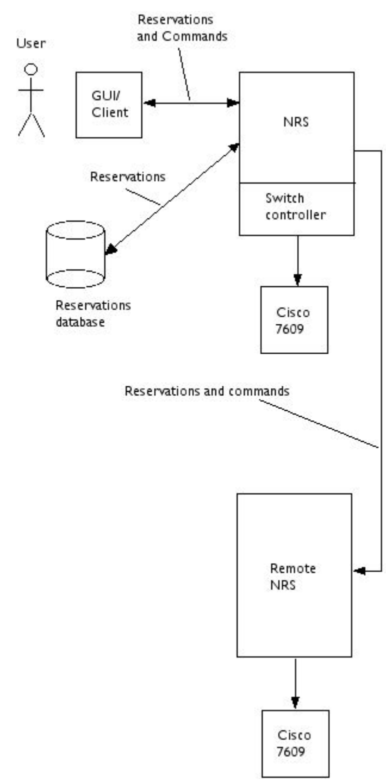

Figure 5: The NRS architecture

5 Current Status

We have implemented a simple GUI-based Java program to configure connections in a Cisco 7609

[12]See URL: `http://www.postgresql.org/`

[13]See URL: `http://www.beepcore.org/`

[14]See URL: `http://egee-jra4.web.cern.ch/EGEE-JRA4/`

in real-time. More importantly, we have converted the basic NRS to create switched-circuit connections. Thus, we can create switched circuits between ESLEA end-hosts placed around the periphery of UKLight (see Figure 1). This software will be demonstrated at the meeting. Interacting with a simulated core switch (Figure 2) is not yet available.

6 Future Work

Likely future enhancements include the following.

1. Add an interface to allow interaction with a network core which can provide switching.
2. Replace the Beep protocol with conventional Web services as the mechanism for communication between NRS components. Though the Beep protocol is a standard it is little used, whereas Web services are widely available. We will probably use Axis[15] from the Apache Software Foundation[16].
3. We have been tracking the security architecture being developed as part of the EGEE BAR work. It has much in common with the security requirements for the CPS and we may well adopt it as a replacement for NRS' own security.
4. Improve the user interface and allow more flexible and abstract ways of specifying reservations.
5. The NRS approach of controlling the periphery of a network and communicating with the core via an interface seems a realistic one which is likely to prove important. We hope to investigate how such a model can interact with emerging ways of switching optical networks, such as GMPLS (General Multiprotocol Label Switching).[17]

Acknowledgements

We are grateful to Saleem Bhatti and Richard Smith for useful discussions about NRS and to Charaka Palansuriya, Kostas Kavoussanakis and Alistair Phipps for useful discussions about the EGEE BAR effort. ESLEA is funded by the Engineering and Physical Science Research Council, the Medical Research Council and the Particle Physics and Astronomy Research Council.

[15]See URL: `http://ws.apache.org/axis/`
[16]See URL: `http://www.apache.org/`
[17]See URL: `http://www.polarisnetworks.com/gmpls/`

References

[1] C. Greenwood, V. Bartsch, P. Clarke, P. Coveney, C. Davenhall, B. Davies, M. Dunmore, B. Garrett, M. Handley, R. Hughes-Jones, R. Jones, M. Lancaster, L. Momtahan, S. Pickles, R. Pinning, A. Simpson, R. Spencer and R. Tasker, 2005, 'Exploitation of Switched Light Paths for e-Science Applications (ESLEA)', these proceedings.

[2] M. Rio, A. di Donato, F. Saka, N. Pezzi, R. Smith, S. Bhatti and P. Clarke, 2003, 'Quality of Service Networking for High Performance Grid Applications', *Journal of Grid Computing* (Springer Science + Business Media B.V, Formerly Kluwer Academic Publishers B.V.), **1**, no. 4, pp329-343.

[3] A. Patil, M. Buchli, B. Vial, A. Di Donato and P. Clarke, 2004, 'Specification of Interfaces for Bandwidth Reservation Service' (EGEE-DJRA4.1-501154-v1.0.doc).