



034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds



Deliverable reference number: D.2.10

Final Grid-GMPLS Control Plane prototype

Due date of deliverable: 2008-12-31
Actual submission date: 2008-12-31
Document code: Phosphorus-WP2-D2.10

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable: **Nextworks (NXW)**

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level

PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	



Final Grid-GMPLS Control Plane prototype

Abstract

This document contains the release notes for the final G²MPLS Network Control Plane prototype. It consists of all the software modules designed, implemented and publicly demonstrated by the Phosphorus WP2. The prototype includes the PEPGW module for the integration with the WP4 AuthN/AuthZ framework, the gateway module between G²MPLS Control Plane and WP1-Harmony, plus some refinements and fixings to the previously released G²MPLS prototype (D2.5) that derived from the G²MPLS consolidation activity.

This deliverable constitutes the official and public release of G²MPLS Control Plane in the form of release notes and software package.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



List of Contributors/Software Developers

Giacomo Bernini	NXW
Gino Carrozzo	NXW
Nicola Ciulli	NXW
Giodi Giorgi	NXW
Francesco Salvestrini	NXW
Giada Landi	NXW
Jakub Gutkowski	PSNC
Adam Kaliszan	PSNC
Lukasz Lopatowski	PSNC
Damian Parniewicz	PSNC
Eduard Escalona	UESSEX

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Table of Contents

0	Executive Summary	6
1	Objectives and Scope	7
2	Terminology	8
3	G ² MPLS Control Plane final prototype	9
3.1	Package format	9
3.2	Package contents	10
3.3	Start-up and shut-down procedures	11
3.3.1	Single protocols	11
3.3.2	Python modules	12
3.3.3	G ² MPLS Controllers	13
4	G ² MPLS prototype configuration	16
4.1	Transport Plane resources	17
4.2	Control Plane logical topology	18
5	Demonstrations of G ² MPLS Control Plane	21
5.1	G ² MPLS test-bed topology	21
5.2	Configurations of the G ² MPLS controllers	22
6	References	26
7	Acronyms	27

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



List of Figures

Figure 3-1: Phosphorus G ² MPLS prototype build structure.	11
Figure 5-1: Phosphorus G ² MPLS testbed: physical layout.	23
Figure 5-2: Phosphorus G ² MPLS testbed: SCN topology.	24
Figure 5-3: Phosphorus G ² MPLS testbed: Control Plane logical topology	25

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



0 Executive Summary

This document provides the release notes of the final G²MPLS Network Control Plane prototype delivered in the form of a XEN virtual machine.

In section 3 the contents and basics for operations of the G²MPLS prototypes are described for the different deployment cases: G²MPLS core controller, G²MPLS border controller (i.e. with G.E-NNI), G²MPLS edge controller (i.e. with G.UNI-N) and G²MPLS G.UNI client node (i.e. with guni-gw functionality towards the Grid middleware).

In section 4 the configuration process is detailed, in order to set a general reference for any users in his specific deployment scenario.

In Section 5 some details on the topology and configuration of the G²MPLS Control Plane in the multi-domain heterogeneous Phosphorus test-bed are provided. This Control and Transport Plane layouts have been used in the public G²MPLS demonstrations and experiments during Supercomputing 2008 (Austin-TX, USA on 16-21 November 2008) and the ICT'08 (Lyon-FR, EU, on 25-27 November 2008).

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



1 Objectives and Scope

This document briefly describes the contents of the G²MPLS software package. The detailed architectural background and system design notes have been provided in the other WP2 deliverables, listed below:

- D2.1 “The Grid-GMPLS Control Plane architecture”,
- D2.2 “Routing and Signalling Extensions for the Grid-GMPLS Control Plane”
- D2.7 “Grid-GMPLS network interfaces specification”
- D2.3 “Grid-GMPLS high level system design”
- D2.6 “Deployment models and solutions of the Grid-GMPLS Control Plane”

Basic configuration hints and bootstrap procedures are the core of this document, in order to provide a general reference for any users willing to deploy G²MPLS in its own Transport Network with a specific SCN, addressing spaces and interconnections between equipments. Detailed explanations on the available configuration commands exposed by each software module can be retrieved during operation of the stack, by logging into the VTY interface and using the module help.

Most of the G²MPLS modules released in this prototype are based on the QUAGGA routing suite, and therefore the QUAGGA official documentation [QUAGGA-DOC] complements these notes.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



2 Terminology

No specific terminology is introduced by this document, which refers to Deliverable D2.1, D2.2, D2.6, D2.7 and D2.4 for any specific terms used.

A full list of the abbreviations used in this document is provided in Section 7.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



3 G²MPLS Control Plane final prototype

3.1 Package format

The final G²MPLS prototype is released in the form of a XEN virtual machine, configured with all the system packages (libraries and programs) needed for the correct operations of the G²MPLS software modules.

The G²MPLS virtual machine is a XEN Domain U (DomU) based on a Linux/Gentoo distribution for x86 32-bit platforms. It is built with XEN capabilities activated in its kernel 2.6.16. The hosting server from which it has been derived (XEN Domain 0 – Dom0) is a Linux/Ubuntu 7.04 with kernel 2.6.19 and XEN 3.0 installed.

The G²MPLS XEN VM consists of two disks

- g2mpls_controller.sda1, containing the system root (“/”)
- g2mpls_controller_swap.sda2, representing the swap memory for that system

The XEN dom0 configuration which is needed to start the G²MPLS XEN VM is provided in the following excerpt to be added as independent file in `/etc/xen/seeds` directory.

```
kernel = "/mnt/xen/vmlinuz-2.6.16-xenU"
memory = 128
name = " g2mpls_controller "
disk = ['file:/mnt/xen/seeds/ g2mpls_controller.sda1,sda1,w', 'file:/mnt/xen/seeds/
g2mpls_controller_swap.sda2,sda2,w']
root = "/dev/sda1 ro"
vif = []
cpus = "0-1"
vcpus = 2
```

Code 3-1: G²MPLS XEN VM configuration file in dom0.

The G²MPLS VM boots with pre-configured hostname (`g2mpls-controller`) and IP address for its virtual network interface. Both can be overridden in `/etc/conf.d` according to the user's needs. The root user can be accessed with password “g2mpls”.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



3.2 Package contents

The G²MPLS prototype comes up with a preconfigured user (user01, password user01) and the main object codes of G²MPLS software modules, as listed in Figure 3-1. These components are contained in the directory `phosphorus-g2mpls` located in `/home/user01`. In details:

- `phosphorus-g2mpls/build` contains the protocols executables and the common libraries of the G²MPLS stack; specifically
 - `phosphorus-g2mpls/build/sbin` groups most of the G²MPLS protocols executables ([PH-WP2-D2.3], [PH-WP2-D2.9]);
 - `phosphorus-g2mpls/build/pyg2mpls` groups all the python components (NCC, CCC and framework tools) just described in [PH-WP2-D2.3];
 - `phosphorus-g2mpls/build/etc` contains the controllers run-scripts and some minimal example of configurations (more detailed configurations are provided in G²MPLS-DEMOS folder);
 - `phosphorus-g2mpls/build/pepgw` groups all the PEPGW components and the GAAATK released by WP4 for AAA integration [PH-WP2-D2.8]
- `phosphorus-g2mpls/G2MPLS-DEMOS` contains all the configurations of the G²MPLS controllers used during the G²MPLS demonstrations in SC'08 and ICT'08 conferences and exhibitions (ref. section 5 for a topology description).

The G²MPLS prototype is based on the Quagga v0.99.7 substrate [QUAGGA-DOC] from which it inherits the base OSPFv2 implementation and some common libraries and tools. Many other functionalities and protocols are implemented in the form of independent processes, also based on the QUAGGA framework. Therefore, most of the G²MPLS modules/processes expose a VTY interface for the inspection and configuration and it is similar to the command line interfaces of the other QUAGGA protocols.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10

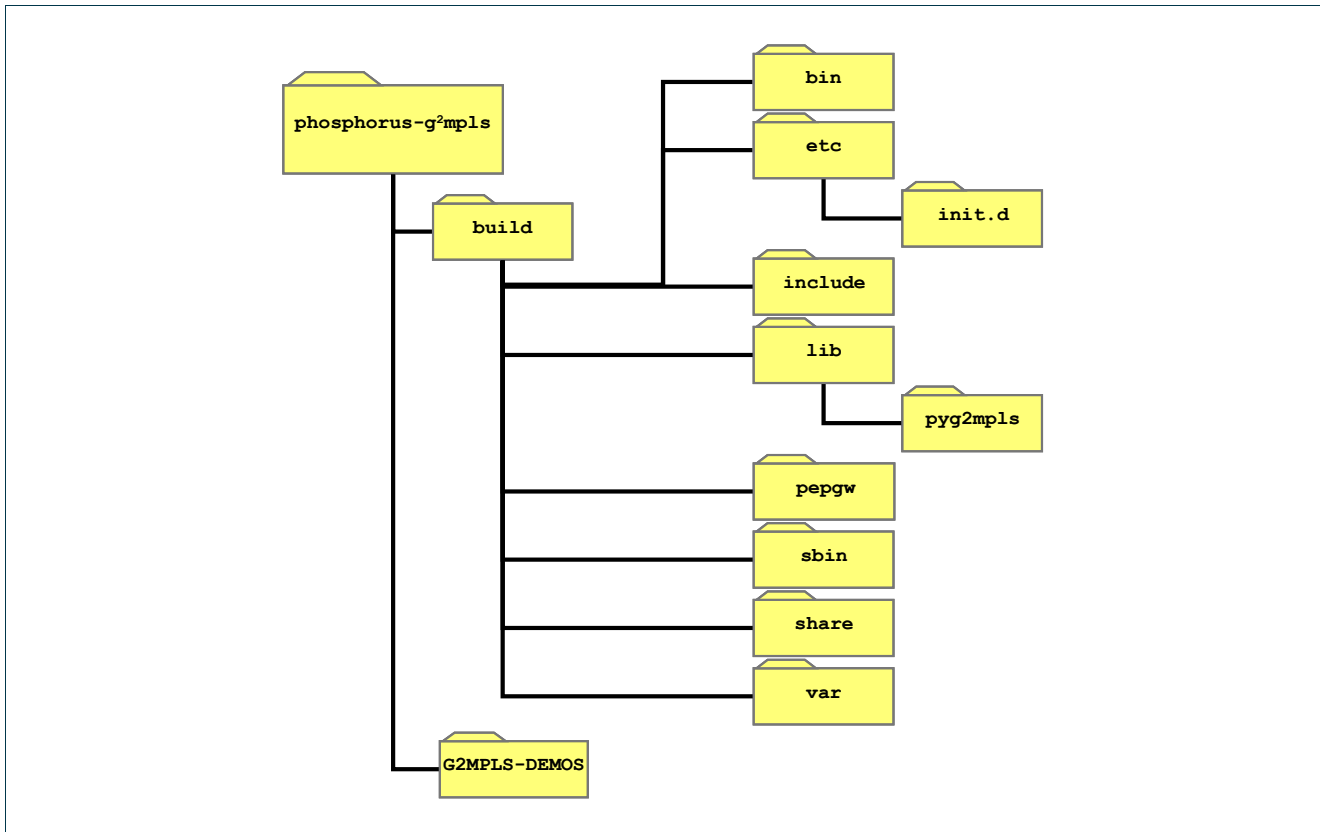


Figure 3-1: Phosphorus G²MPLS prototype build structure.

3.3 Start-up and shut-down procedures

3.3.1 Single protocols

NOTE. This procedure is deprecated for a full controller operation, since most of the protocols depend on the existence of other modules and open CORBA interfaces. These dependencies are preserved by the init scripts for the full G²MPLS controllers (core, border, edge, uni-client) described in sections

Each process in `./build/sbin` can be run with a set of options, briefly described below.

```
"Usage : PROGRAM-NAME [OPTION...]"
    "Daemon which manages PROGRAM-NAME module"

    -d, --daemon           Runs in daemon mode"
    -f, --config_file     Set configuration file name"
    -i, --pid_file        Set process identifier file name"
    -C, --dryrun          Check configuration and exit"
    -o, --iors_dir        Set IORs directory"
```

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

```
"-P, --vty_port    Set vty's port number"  
"-u, --user        User to run as"  
"-g, --group       Group to run as"  
"-v, --version     Print program version"  
"-h, --help       Display this help and exit"
```

Code 3-2: G²MPLS process run options.

Automated bash scripts have been provided in `build/etc/init.d` to start-up, shut-down and restart singularly most of these executables. They locate the configuration files, set some options and run/kill the G²MPLS modules.

Each G²MPLS daemon can be checked for its correct operation and possibly further configured through its VTU, which is a command line interface accessed via telnet to the `g2mpls-controller` at the ports specified below.

```
TNRCD_VTY_PORT      2613  
LRMD_VTY_PORT       2610  
SCNGWSD_VTY_PORT    2620  
OSPF_VTY_PORT       2604  
G2RSVPTED_VTY_PORT 2630  
GENNI_G2RSVPTED_VTY_PORT 2631  
GUNIN_G2RSVPTED_VTY_PORT 2632  
GUNIC_G2RSVPTED_VTY_PORT 2633  
G2PCERAD_VTY_PORT   2615  
GUNIGWD_VTY_PORT    2614  
HG2GWD_VTY_PORT     2625  
NCCD_VTY_PORT       2616
```

Code 3-3: G²MPLS main VTU ports.

The configuration file specified as a run option of each daemon contains the VTU commands that are read at the bootstrap of the protocol. Therefore, a set of the available commands per protocol can be inferred by these files or retrieved exhaustively through the help of the VTU interface.

3.3.2 Python modules

NOTE. This procedure is deprecated for a full controller operation, since most of the protocols depend on the existence of other modules and open CORBA interfaces. These dependencies are preserved by the init scripts for the full G²MPLS controllers (core, border, edge, uni-client) described in sections

The binaries for NCC and CCC modules are located in `./build/lib/pyg2mpls`, which is linked by the python site-packages location on the VM (`/usr/lib/python.x.x/site-packages/`). The start-up/shut-down procedures for these modules are wrapped by a bash script:

```
`build/lib/tools/pyrun.sh build/lib/tools/pyg2` <action> <protocol>  
  <action>      := { start | stop | restart }  
  <protocol>    := { nccd | cccd | rcd }
```

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

```
<instance_no> := number of protocol instances to be run on the same controller  
                (just for the single-module debugging purposes)
```

Code 3-4: G²MPLS Python objects run script.

NOTE. the execution of any xCC modules and of RCD generates persistency files (*.pdb) that must be removed for a correct restart from scratch of the modules.

All the activities of these python modules depend on actions from G²MPLS protocols or CORBA interfaces. Therefore, the VTY interface they export is a stand-alone process, the nccd, and is mainly used for show commands and creation of Soft Permanent Connections (SPC).

3.3.3 G²MPLS Controllers

NOTE. In order to simplify all the operations, it is suggested to run the controllers described below from the root user. This is necessary for those G²MPLS protocols using the system sockets.

Reference init bundle scripts are

```
build/etc/init.d/g2mpls-core-ctrl  
  
build//etc/init.d/g2mpls-border-ctrl  
  
build//etc/init.d/g2mpls-edge-ctrl  
  
build//etc/init.d/g2mpls-uniclient-ctrl
```

3.3.3.1 G²MPLS core controller

The modules needed by a G²MPLS core controller are:

- tnrcd, i.e. the process in charge of implementing the mediation between Control Plane and the Transport Plane equipment;
- lrmd, i.e. the process storing the Control Plane data model and the internal bindings between resources
- scngwsd, i.e. the process that bridges the set of SCN interfaces with the TE-Links and related Control Channels;
- g2rsvptd, i.e. the process implementing the G.I-NNI G2.RSVP-TE;
- ospfd, i.e. the process implementing the I-NNI G2.OSPF-TE;
- g2pcerad, i.e. the process implementing the routing algorithms on the G²MPLS multi-domain topologies.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

A G²MPLS core controller can be run with the init script `build/etc/init.d/g2mpls-core-ctrl` in which the correct launch sequence is preserved across different start/stop/restart events.

NOTE. In case of failure of any G²MPLS daemons a full restart of the controller must be done, due to the lack of process dependency tracking in this preliminary release.

3.3.3.2 G²MPLS border controller

The modules needed by a G²MPLS border controller are:

- `tnrcd`;
- `lrmd`;
- `scngwsd`;
- `pepgw`
- the python `nccd` and `rcd`
- `g2rsvpted` implementing the G.I-NNI G2.RSVP-TE;
- `g2rsvpted` implementing the G.E-NNI G2.RSVP-TE;
- `ospfd`, i.e. the process implementing the I-NNI G2.OSPF-TE;
- `g2pcerad`, i.e. the process implementing the routing algorithms on the G²MPLS multi-domain topologies.
- `nccd`, i.e. the process implementing the VTY interface to the python xCCs modules

A G²MPLS border controller can be run with the init script `build/etc/init.d/g2mpls-border-ctrl` in which the correct launch sequence is preserved across different start/stop/restart events.

NOTE. In case of failure of any G²MPLS daemons a full restart of the controller must be done, due to the lack of process dependency tracking in this preliminary release.

3.3.3.3 G²MPLS edge controller

The modules needed by a G²MPLS edge controller are:

- `tnrcd`;
- `lrmd`;
- `scngwsd`;
- `pepgw`
- the python `nccd` and `rcd`
- `g2rsvpted` implementing the G.I-NNI G2.RSVP-TE;
- `g2rsvpted` implementing the G.UNI-N G2.RSVP-TE;
- `ospfd`, i.e. the process implementing the I-NNI G2.OSPF-TE and the UNI flooding of Grid information;

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

- g2pcerad, i.e. the process implementing the routing algorithms on the G²MPLS multi-domain topologies.
- nccd, i.e. the process implementing the VTY interface to the python xCCs modules

A G²MPLS edge controller can be run with the init script `build/etc/init.d/g2mpls-edge-ctrl` in which the correct launch sequence is preserved across different start/stop/restart events.

NOTE. In case of failure of any G²MPLS daemons a full restart of the controller must be done, due to the lack of process dependency tracking in this preliminary release.

3.3.3.4 G²MPLS G.UNI CLIENT controller

The modules needed by a G²MPLS G.UNI CLIENT controller are:

- tnrcd, always using a simulator for the Transport Plane;
- lrmd;
- scngwsd;
- pepgw
- the python ccd
- g2rsvptd implementing the G.UNI-C G2.RSVP-TE;
- ospfd, i.e. the process implementing the UNI flooding of Grid information;
- gunigwd, i.e. the process implementing the gateway functionality between the G²MPLS protocols and the WS-Agreement interface towards the Grid Middleware.
- the python g2dialer, i.e. an additional module implementing a direct bridging between the application layer (in particular DDSS) and the G²MPLS protocols.

A G²MPLS G.UNI CLIENT controller can be run with the init script `build/etc/init.d/g2mpls-uniclient-ctrl` in which the correct launch sequence is preserved across different start/stop/restart events.

NOTE. In case of failure of any G²MPLS daemons a full restart of the controller must be done, due to the lack of process dependency tracking in this preliminary release.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



4 G²MPLS prototype configuration

The configuration files distributed with the G²MPLS prototype represent a simple and not exhaustive reference, released just to let the controller boot and start its operation. A user should customize those configurations and possibly extend them in order to fit his choices about address spaces, functionalities and Control Plane scenarios to implement.

Most of the Control Plane configurations are contained in the configuration files of tnrzd and lrmz. lrmz acts a hub element for all the protocols, in particular the ospfd and the g2rsvpted. For this reason, the launch sequence of the daemons is crucial and must follow the order:

1. tnrzd
2. lrmz
3. scngwd
4. pepgw (where needed, ref. [PH-WP2-D2.8] for detailed scenarios)
5. Python Object Codes
6. g2rsvpted
7. ospfd
8. g2pcerad
9. nccd

Only this order can guarantee the openness of the needed CORBA interfaces and the meaningfulness of the data model information, as explained in sections 3.3.3.1, 3.3.3.2, 3.3.3.3 and 3.3.3.4.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



4.1 Transport Plane resources

The Transport Plane data model is stored in TNRC. The `tnrcd.conf` file specifies the location of the equipment. This equipment could be a real Transport Plane equipment or a simulator. In the first case, the `tnrcd` automatically retrieves TN resources from the equipment, while in the latter case (simulator) ports and labels must be declared by the user. The simulator option can be very useful to run the G²MPLS Control Plane without connecting to real hardware, a feature that can help the training/learning or the pre-production phases.

Examples of `tnrcd` configuration files for the simulator are provided in the following:

```
!*****NODE 1 - EQUIPMENT CONFIGURATION FILE*****!  
!  
! Enter in TNRC_NODE  
tnrc  
!  
! EQUIPMENT 1  
  
eqpt id 1 addr 1.1.1.1 type simulator opstate up admstate enabled location XXX  
  
! BOARD 1  
board id 1 eqpt-id 1 sw-cap fsc enc-type fiber opstate up admstate enabled  
  
! PORT 0x1101  
port id 0x1101 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 1.1.1.2 rem-portid 1 opstate  
up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base 0 lambdas-count  
0  
  
! PORT 0x1201  
port id 0x1201 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 1.1.1.2 rem-portid 0x2101  
opstate up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base 0  
lambdas-count 0  
  
! PORT 0x1301  
port id 0x1301 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 1.1.1.2 rem-portid 0x3101  
opstate up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base 0  
lambdas-count 0  
  
! PORT 0x1401  
port id 0x1401 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 1.1.1.2 rem-portid 0x4101  
opstate up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base 0  
lambdas-count 0
```

Code 4-1: A sample `tnrcd` simulator file for Fiber Switching Capable equipments.

```
!*****NODE 1 - EQUIPMENT SIMULATOR CONFIGURATION FILE*****!!  
!  
! Enter in TNRC_NODE  
tnrc  
!  
! EQUIPMENT 1  
  
eqpt id 1 addr 150.254.212.137 type simulator opstate up admstate enabled location XXX  
  
! BOARD 1
```

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

```
board id 1 eqpt-id 1 sw-cap lsc enc-type lambda opstate up admstate enabled

! PORT 0x1108
port id 4360 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 150.254.212.139 rem-portid 4365
opstate up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base
0x2900000b lambdas-count 40
resource id 0x24 port-id 4360 board-id 1 eqpt-id 1 type lsc tp-flags 0 opstate up
admstate enabled state free

! PORT 0x110D
port id 4365 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 150.254.212.138 rem-portid 4360
opstate up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base
0x2900000b lambdas-count 40
resource id 0x24 port-id 4365 board-id 1 eqpt-id 1 type lsc tp-flags 0 opstate up
admstate enabled state free

! PORT 0x110E
port id 4366 board-id 1 eqpt-id 1 flags 0 rem-eq-addr 171.16.50.1 rem-portid 1 opstate
up admstate enabled bw 0x4A99CE3D protection unprotected lambdas-base 0x2900000b
lambdas-count 40
resource id 0x24 port-id 4366 board-id 1 eqpt-id 1 type lsc tp-flags 0 opstate up
admstate enabled state free
```

Code 4-2: A sample tnrcd simulator file for Lambda Switching Capable equipments.

4.2 Control Plane logical topology

The Control Plane resources are all maintained by the LRM. In `lrmd.conf` the Control Plane logical topology is detailed in terms of:

- router ID of the `g2mps_controller`
- SCN interfaces used to receive/transmit protocol packets
- Control Channels
- TE-links with their TE attributes (adjacency type, TE metric, colours, SRLGs, TNAs, etc.)
- Data-links (in 1:1 correspondence with those loaded by `tnrcd` and exported at the TNRC's CORBA interface)
- bindings of TE-links with Control Channels
- insertion of Data-links into TE-links.

This information is centralized and used by all the protocols for routing and signalling. Therefore, the `lrmd` configuration file is the larger than the configuration files of the upper protocols, which inherit most of the information from it.

An example of more complex LRM configuration is provided in the following excerpt.

```
!*****NODE 1 - LRM CONFIGURATION FILE*****!
!
!
```

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

```
hostname ADVA1_lrmd
password zebra
enable password zebra

log file /home/user01/phosphorus-g2mpls/build/var/lrmd.log

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! User specific part (add your commands from here)
!
! starting LRM and entering its context
lrmd
!
router-id 192.168.100.137
router-uni-id 192.168.150.137
!
! INNI control interface
scn-if add ip 192.168.100.137
scn-if en ip 192.168.100.137
!
! UNI control interface
scn-if add ip 192.168.150.137
scn-if en ip 192.168.150.137
!
!
! Adva1 - Adva2
cc add ccid 0x12 scn-ip 192.168.100.137 scn-nbr 192.168.100.138
cc en ccid 0x12
cc up ccid 0x12
!
! Adva1 - Adva3
cc add ccid 0x13 scn-ip 192.168.100.137 scn-nbr 192.168.100.139
cc en ccid 0x13
cc up ccid 0x13
!
! Adva1 TNA
cc add ccid 0x1101 scn-ip 192.168.150.137 scn-nbr 192.168.150.115
cc en ccid 0x1101
cc up ccid 0x1101
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! TELink Adva1 - Adva2 !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
te-link-inni add ipv4 1.2.1.1 pref-len 30 nbr-id 192.168.100.138 rem-telink-id 1.2.1.2
te-link bind ipv4 1.2.1.1 pref-len 30 ccid 0x12
te-link en ipv4 1.2.1.1 pref-len 30
!
!! DataLinks !!
data-link add id 0x04011108 nbr-id 0x0401110D
data-link en id 0x04011108
!
!
te-link push ipv4 1.2.1.1 pref-len 30 data-link-id 0x04011108
te-link set ipv4 1.2.1.1 pref-len 30 tem 100
te-link run ipv4 1.2.1.1 pref-len 30
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! TELink Adva1 - Adva3 !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

```
te-link-inni add ipv4 1.3.1.1 pref-len 30 nbr-id 192.168.100.139 rem-telink-id 1.3.1.2
te-link bind ipv4 1.3.1.1 pref-len 30 ccid 0x13
te-link en ipv4 1.3.1.1 pref-len 30
!
!! DataLinks !!
data-link add id 0x0401110D nbr-id 0x04011108
data-link en id 0x0401110D
!
te-link push ipv4 1.3.1.1 pref-len 30 data-link-id 0x0401110D
te-link set ipv4 1.3.1.1 pref-len 30 tem 100
te-link run ipv4 1.3.1.1 pref-len 30
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!! TELink TNA !!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
te-link-uni add ipv4 1.101.1.1 pref-len 30 nbr-id 192.168.150.115 rem-telink-id
1.101.1.2
te-link set ipv4 1.101.1.1 pref-len 30 tna 10.10.10.1 pref-len 24
te-link bind ipv4 1.101.1.1 pref-len 30 ccid 0x1101
te-link en ipv4 1.101.1.1 pref-len 30
!
!! DataLink !!
data-link add id 0x0401110E nbr-id 0x04010001
data-link en id 0x0401110E
!
te-link push ipv4 1.101.1.1 pref-len 30 data-link-id 0x0401110E
te-link run ipv4 1.101.1.1 pref-len 30
```

Code 4-3: A sample Irmcd configuration file.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



5 Demonstrations of G²MPLS Control Plane

This chapter presents the multi-domain heterogeneous optical test-bed using the G²MPLS Control Plane that has been used in the public demonstrations and experiments during Supercomputing 2008 (Austin-TX, USA on 16-21 November 2008) and the ICT'08 (Lyon-FR, EU, on 25-27 November 2008).

The demonstrations have been focused on the DDSS application for remote anycast storage of data contents. This application built in the framework of Phosphorus WP3 has been interfaced to the G²MPLS Control Plane, and the Phosphorus Integrated model with anycasting has been successfully demonstrated (ref. [PH-WP2-D2.1]).

5.1 G²MPLS test-bed topology

The G²MPLS Control Plane has been run in two local test-beds, PSNC-PIONIER (Poland) and UESSEX (UK), interconnected through the GÉANT2 network (ref. Figure 5-1). Two types of switching capabilities have been demonstrated in the two administrative domains:

- the Fiber Switching Capability (FSC) in UESSEX test-bed, built around one Calient Diamond Wave Fiber Connect equipment partitioned into four independent fiber-switching sub-nodes for Control Plane purposes. Each sub-switch has been interconnected with bi-directional optical fibers to the other sub-switches to realize a fully meshed topology. Each sub-switch has been controlled by a G²MPLS controller running the appropriate instance of G²MPLS Control Plane (edge, core or border node). The four controllers have been connected through a dedicated 1GbE local area network, which constitutes the Signaling Control Network (SCN) for the G²MPLS Control Plane.
- the Lambda Switching Capability (LSC) in PSNC-PIONIER test-bed, based on an optical DWDM ring with three ADVA FSP 3000RE-II ROADMs. These equipments have been inter-connected with bi-directional optical fibers and the DWDM system configured to hold up to 40 wavelengths. Each ROADM has been controlled by a G²MPLS controller running the appropriate instance of G²MPLS Control Plane (edge, core or border node). These three controllers have been connected through a dedicated 1GbE local area network implementing the SCN.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

The two test-beds have been completed with the respective G.UNI clients towards the application/Grid layer equipments: one client in PSNC, another client and two servers in UESSEX.

The interconnection between the two test-beds has been implemented through VLAN switching on a GÉANT2 connection between PSNC and UESSEX laboratories (ref. WP6 topologies for details). Each inter-domain TE-link has been mapped into a specific VLAN, as well as the SCN which has been mapped into another VLAN. These configurations have been treated completely opaque at G²MPLS level and under the control of WP6 test-bed operators.

The final SCN topology resulting from this process is depicted in Figure 5-2.

5.2 Configurations of the G²MPLS controllers

Different types of G²MPLS controllers have been demonstrated in the Phosphorus testbed, according to the logical topology shown in Figure 5-3.

A dedicated folder in the released prototype (`/home/user01/phosphorus-g2mpls/G2MPLS-DEMOS`) contains all the configuration files of the G²MPLS modules run on each node. The purpose of this directory is to provide a reference for G²MPLS users about the commands needed to setup the G²MPLS Control Plane in this complex multi-domain scenario.

An executive description of the G²MPLS capabilities demonstrated in the Phosphorus testbed is provided in the form of a video clip on the Phosphorus website (<http://www.ist-phosphorus.eu/documents.php>).

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

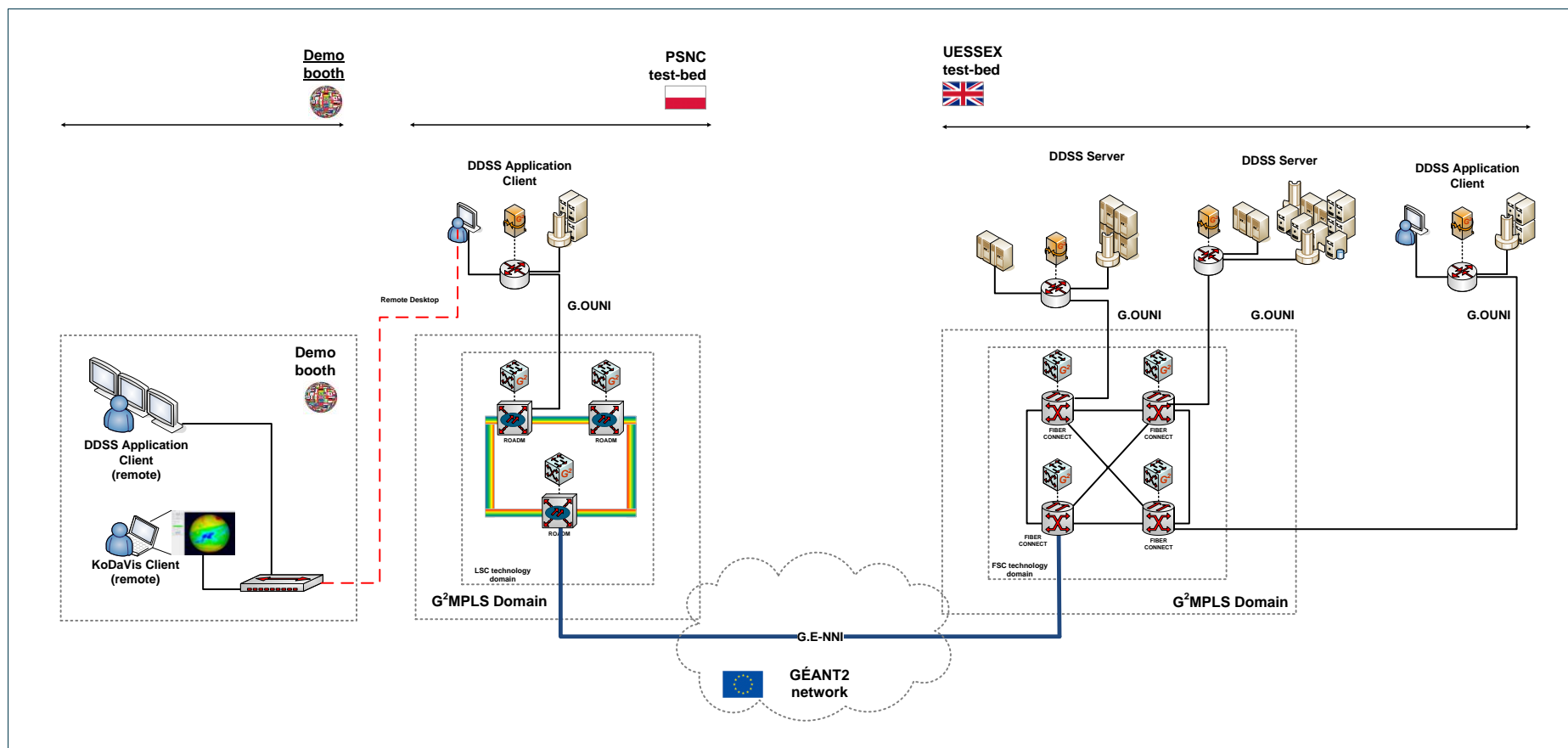


Figure 5-1: Phosphorus G²MPLS testbed: physical layout.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

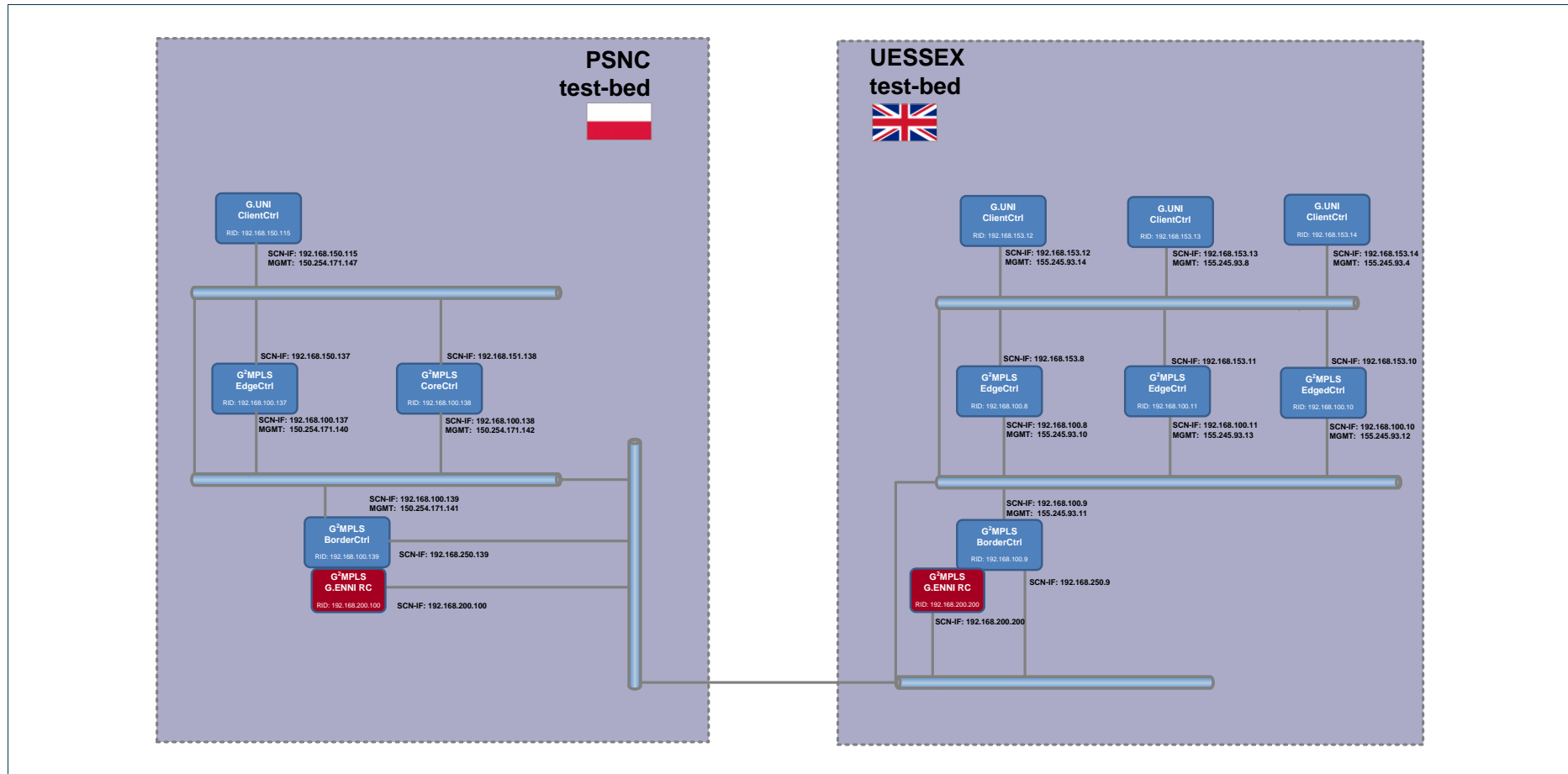


Figure 5-2: Phosphorus G²MPLS testbed: SCN topology.

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

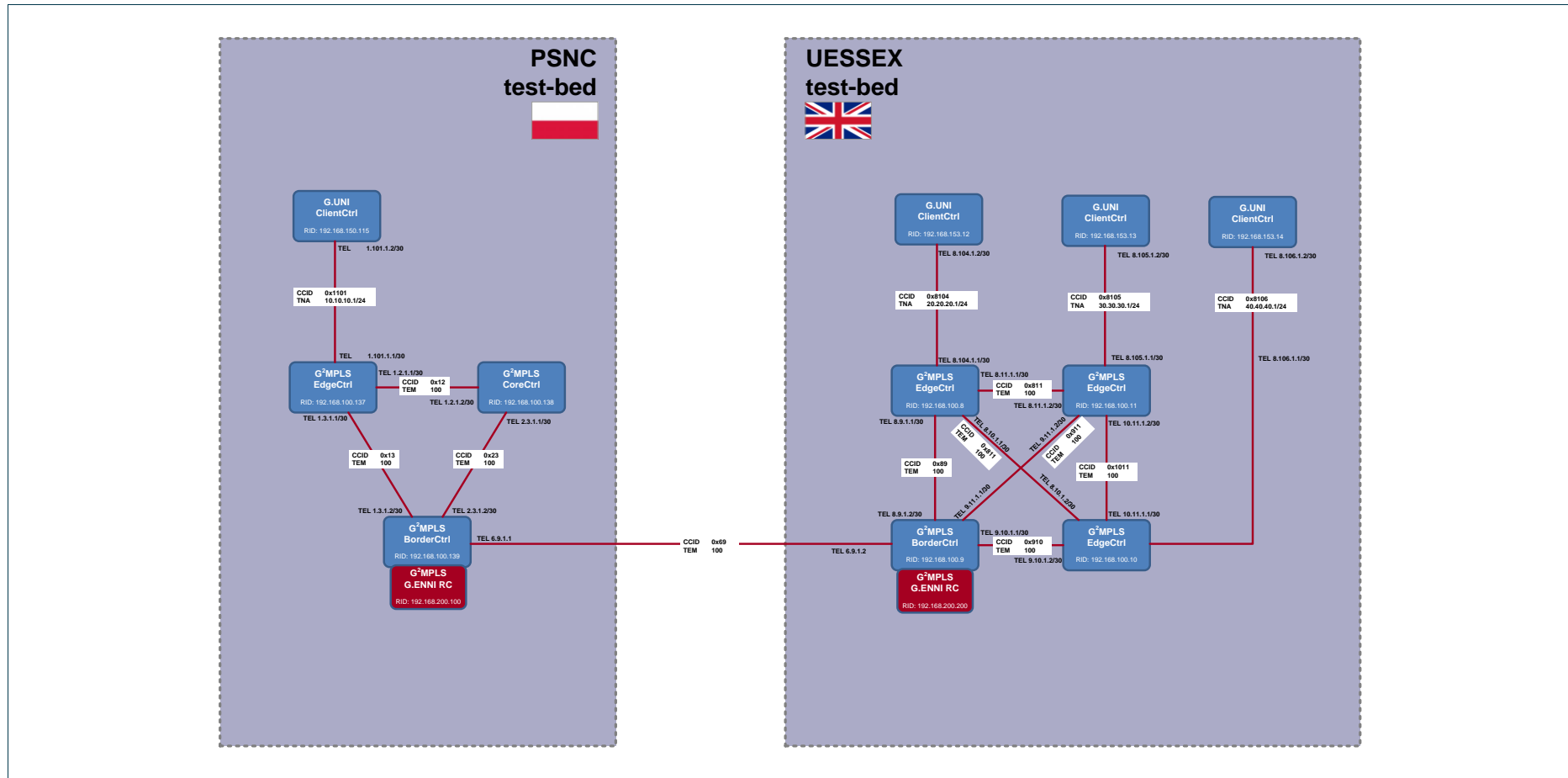


Figure 5-3: Phosphorus G²MPLS testbed: Control Plane logical topology

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



6 References

The references listed here are only those directly functional to this document. For a list of the references to standards appearing in this document, please point to D2.1, D2.2, D2.3, D2.4, D2.6 and D2.7.

[PH-WP2-D2.1]	Phosphorus deliverable D2.1, "The Grid-GMPLS Control Plane architecture".
[PH-WP2-D2.2]	Phosphorus deliverable D2.2, "Routing and Signalling Extensions for the Grid-GMPLS Control Plane".
[PH-WP2-D2.3]	Phosphorus deliverable D2.3, "Grid-GMPLS high level system design".
[PH-WP2-D2.4]	Phosphorus deliverable D2.4, "Report on Grid-GMPLS Control Plane functional tests".
[PH-WP2-D2.5]	Phosphorus deliverable D2.5, "Preliminary Grid-GMPLS Control Plane prototype".
[PH-WP2-D2.6]	Phosphorus deliverable D2.6, "Deployment models and solutions of the Grid-GMPLS Control Plane".
[PH-WP2-D2.7]	Phosphorus deliverable D2.7, "Grid-GMPLS network interfaces".
[PH-WP2-D2.8]	Phosphorus deliverable D2.8, "Design of the Grid-GMPLS Control Plane to support the Phosphorus Grid AAI".
[PH-WP2-D2.9]	Phosphorus deliverable D2.9, "Design of Grid-GMPLS interworking with NRPS".
[QUAGGA-DOC]	The Quagga Software Routing Suite documentation. http://www.quagga.net/docs/docs-info.php
[CORBA]	http://www.corba.org/
[omniORB]	http://omniorb.sourceforge.net/

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



7 Acronyms

AAA	Authentication, Authorisation, and Accounting
AAI	Authentication and Authorization Infrastructure
ANSI	American National Standards Institute
API	Application Programming Interface
ARGON	Allocation and Reservations in Grid-enabled Optical Networks
ASON	Automatically Switched Optical Network
BB	Bandwidth Broker
BGRP	Border Gateway Reservation Protocol
BoD	Bandwidth on Demand
BR	Border Router
CE	Computing Element
CIM	Computer Integrated Manufacturing
COPS	Common Open Policy Protocol
CORBA	Common Object Request Broker Architecture
CP	Control Plane
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CR-LDP	Constraint-based Label Distribution Protocol
DCM	Distributed Call and Connection Management
DCN	Data Communication Network
DRAC	Dynamic Resource Allocation Controller
DVB	Digital Video Broadcasting
DWDM	Dense Wavelength Division Multiplexing
EGEE	Enabling Grids for E-science
EC	European Commission
EMS	Execution Management Services
E-NNI	Exterior NNI
ERO	Explicit Route Object
ETSI	European Telecommunications Standards Institute
EU	European Union
FCAPS	Fault, Configuration, Accounting, Performance, Security

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

G.CR-LDP	G ² MPLS CR-LDP
G.OSPF-TE	GMPLS OSPF-TE
G.OUNI	Grid OUNI
G.OUNI-C	G.OUNI - Client
G.OUNI-N	G.OUNI - Network
G.RSVP-TE	GMPLS RSVP-TE
G²MPLS	Grid-GMPLS (enhancements to GMPLS for Grid support)
GE	Gigabit Ethernet
GÉANT	Pan-European Gigabit Research Network
GGF	Global Grid Forum
GHPN	Grid High Performance Networking
GIS	Grid Information Service
GLUE	Grid Laboratory Uniform Environment
GMPLS	Generalized MPLS
GNS	Grid Network Service
GRAM	Grid Resource Allocation and Management
GSMP	General Switch Management Protocol
HW	Hardware
IANA	Internet Assigned Numbers Authority
IDM	GÉANT2 Inter-domain Manager
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
I-NNI	Interior NNI
IP	Internet Protocol
IPR	Intellectual Property Right
IPSec	IP security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IS-IS	Intermediate System to Intermediate System
ITU	International Telecommunication Union
JSDL	Job Submission Description Language
LAN	Local Area Network
LDP	Label Distribution Protocol
LRMS	Local Resource Management System
LSA	Link State Advertisement
LSDB	Link State Database
LSP	Label Switched Path
LSR	Label Switch Router
MAC	Media Access Control
MAN	Metropolitan Area Network
MP	Management Plane

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

MPLS	Multi Protocol Label Switching
MPI	Message Passing Interface
NCP	Network Control Plane
NJS	Network Job Supervisor
NMS	Network Management System
NNI	Network to Network Interface
NO	Network Operator
NREN	National Research and Education Network
NRPS	Network Resource Provisioning Systems
NSAP	Network Service Access Point
NSP	Network Service Plane
NTP	Network Time Protocol
OAM	Operations, Administration and Maintenance
OGF	Open Grid Forum
OGSA	Open Grid Services Architecture
OIF	Optical Internetworking Forum
OS	Operating System
OSPF	Open Shortest Path First protocol
OSPF-TE	OSPF with Traffic Engineering extensions
O-UNI	Optical UNI
P2MP	Point to Multi Point
PON	Passive Optical Network
POSIX	Portable Operating System Interface
QoS	Quality of Service
RC	Routing Controller
RFC	Request for Comments
RSVP	Resource reSerVation Protocol
RSVP-TE	RSVP with Traffic Engineering extensions
RTP	Real-time Transport Protocol
SDO	Standard Developing Organizations
SE	Storage Element
SLA	Service Level Agreement
SLS	Service Level Specification
SME	Small and Medium Enterprise
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SP	Service Provider
SPF	Sender Policy Framework
SW	Software
TE	Traffic Engineering
TGC	Trusted Computing Group
TL-1	Transaction Language 1
TLS	Transport Layer Security

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10



Final Grid-GMPLS Control Plane prototype

TLV	Type-Length-Value protocol fields
TMF	Tele Management Forum
TN	Transport Network
TO	Telecom Operator
TP	Transport Plane
UCLP	User-Controlled Lightpath Provisioning system
UNI	User to Network Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VLAN	Virtual LAN
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WG	Working Group
WP	Work Package
WS	Web Service
XML	Extensible Markup Language

<END-OF-DOCUMENT>

Project:	Phosphorus
Deliverable Number:	D.2.10
Date of Issue:	31/12/08
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.10