034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number: D1.3 (Prototype)
NRPS southbound interfaces for standard GMPLS control plane

Due date of deliverable: 2007-09-30
Actual submission date: 2007-11-14
Document code: Phosphorus-WP1-D.1.3

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract

This deliverable, named "NRPS southbound interfaces for standard GMPLS control plane", defines the interfaces and methods that have been implemented to provide a GMPLS interface for the Network Resources Provisioning Systems (NRPS) and the Network Service Plane (NSP).

Table of Contents

Table of Figures

0   Executive summary

This deliverable, named "NRPS southbound interfaces for standard GMPLS control plane", defines the interfaces and methods that have been implemented to provide a GMPLS interface for the Network Resources Provisioning Systems (NRPS) and the Network Service Plane (NSP). The interfaces are realized by two modules (cf. Figure 1):

- The GMPLS driver acts as an interface between an NRPS and the GMPLS control plane. It offers a general web service, which is used to create, delete and monitor paths for different GMPLS implementations.

- The Thin NRPS is a network resources provisioning system for domains with a GMPLS control plane. It provides a reservation web service, which is used by the NSP to reserve, create and delete network connections via the GMPLS driver. The Thin NRPS does not need an NRPS adapter, because it has been developed within PHOSPHORUS and therefore can use the NRPS adapter interface, which is common to all PHOSPHORUS NRPS systems.
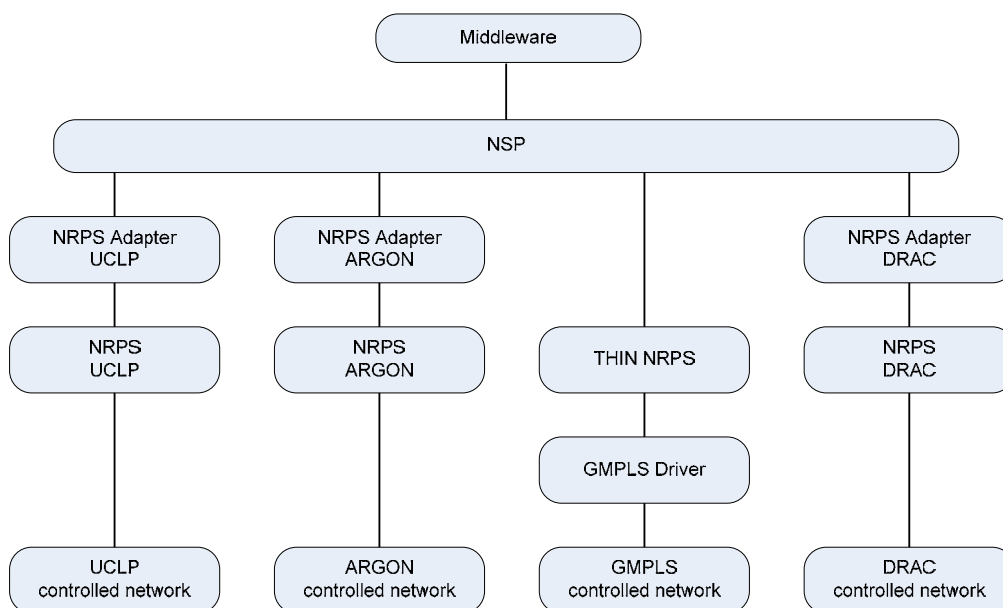
Figure 1: Thin NRPS and GMPLS driver within the PHOSPHORUS context

Within the following chapters the GMPLS driver and the Thin NRPS are described in more detail.

1 GMPLS driver

1.1 GMPLS driver system architecture

The GMPLS driver acts as an interface between an NRPS and the GMPLS control plane. It provides the following services:

Path creation service	This service creates a point-to-point path between two endpoints specified by TNA addresses.
Path termination service	This service tears down a point-to-point path which has been set up by an NRPS.
Path monitoring service	This service provides status information about the specified path.
Path discovery service	This service retrieves any established point-to-point connection in the controlled network. It is basically needed to refresh the NRPS view on the network in case of memory losses through system failures or maintenance reboots.
Endpoint discovery service	This service retrieves information about any endpoint in the controlled network. It is needed to deliver the available endpoints to the NRPS during system initialization and to refresh the NRPS view on the available endpoints in case of memory losses through system failures or maintenance reboots.
Registration service	This service registers the NRPS to receive messages from the web service in case of path status changes or endpoint changes.

Additionally there are notifications which inform registered NRPS about changes concerning paths and endpoints.

Path delete notification	The event driven path delete service informs all registered systems, if a path is no longer available.
Endpoint update notification	The event driven endpoint update service informs all registered systems when endpoints have been removed or added to the controlled network domain.

The overall system design is shown in Figure 2.
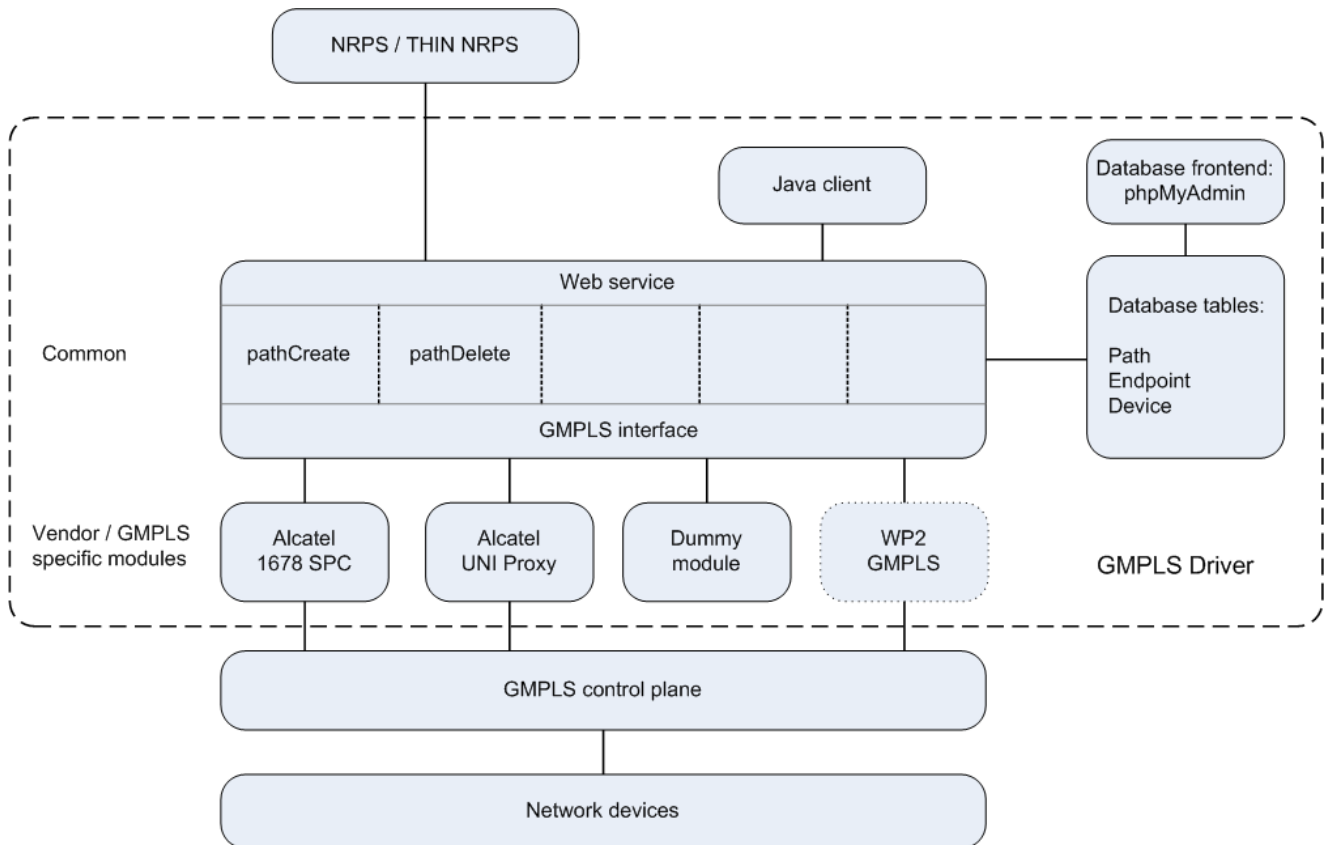
Figure 2: GMPLS driver system design

The GMPLS services are offered via a web service to the clients. A client could be an NRPS or for test purposes a user, who can access the services via a Java application. All Information about paths, endpoints and devices is kept in a MySQL database, which is only accessed by the core component of the GMPLS driver. It can be administered through any database front end like phpMyAdmin. GMPLS operations are initiated and controlled by vendor specific modules. Currently there are three modules available for creating and deleting paths:

-       The Alcatel 1678 SPC module is used for managing the Gigabit Ethernet over SDH and SDH connections within the VIOLA test-bed. This module is based on the Alcatel GMRE command line interface, which allows the creation and deletion of Soft Permanent Connections (SPC).

- The Alcatel UNI Proxy module can be used for signalling Gigabit Ethernet over SDH and SDH connections between two UNI clients. Within the VIOLA test-bed three clients are available, which can be connected to any SDH cross connect.

- The Dummy module does not create any real GMPLS signalled connections. It simulates the creation and deletion of connections within a simulated domain. Besides this, it can act as a sample module for implementing new GMPLS modules.

After the WP2 GMPLS protocol implementation has been finished, a module can be added for this control plane, too.

All GMPLS modules use the same interface towards the GMPLS web service.

Within the following chapters the modules of the GMPLS driver are described in more details.

1.2 GMPLS driver web service

This chapter contains the technical description of the GMPLS web service.

1.2.1 Common types

1.2.1.1 *PathIdentifierType*

PathIdentifier	
Description	Each path can be identified by a unique identifier
XML Type	Int
Java Type	Int
Multiplicity	Once per request
Mandatory	Yes
Details	The identifier is given by an integer

1.2.1.2 *PathType*

SourceTNA	
Description	TNA address of the source port of a path
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	IPv4 address format

DestinationTNA	
Description	TNA address of the destination port of a path
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	IPv4 address format

Bandwidth	
Description	Amount of bandwidth of a path
XML Type	Int
Java Type	Int
Multiplicity	Once per request
Mandatory	Yes
Details	Value specified as Mbps

1.2.1.3 *EndpointType*

EndpointID	
Description	TNA address of the endpoint
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	IPv4 address format

Name	
Description	The name of the endpoint
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	Any string (unique)

Description	
Description	Description of the endpoint
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	No
Details	Description could be used to indicate the name, the port number and location of the device to which the endpoint belongs

Interface	
Description	Interface specifies whether the endpoint is connected to a user or to another domain
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	The following values are currently supported: user, border

DomainId	
Description	Name of the domain to which the endpoint belongs to
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	Any string

Bandwidth	
Description	The bandwidth specifies the maximal bandwidth supported by the endpoint
XML Type	Int
Java Type	Int
Multiplicity	Once per request
Mandatory	Yes
Details	Value specified as Mbps

1.2.2 Services

1.2.2.1 *List of services*

Services	
Path creation service	This service creates a point-to-point path between two endpoints specified by TNA addresses.
Path termination service	This service tears down a point-to-point path which has been set up by an NRPS.
Path monitoring service	This service provides status information about the specified path.
Path discovery service	This service retrieves any established point-to-point connections in the controlled network. It is basically needed to refresh the NRPS view on the network in case of memory losses through system failures or maintenance reboots.
Endpoint discovery service	This service retrieves information about any endpoints in the controlled network. It is needed to deliver the available endpoints to the NRPS during system initialization and to refresh the NRPS view on the available endpoints in case of memory losses through system failures or maintenance reboots.
Registration service	This service registers the NRPS to receive messages from the web service in case of path status changes or endpoint changes.

1.2.2.2 *Creation service (createPath)*

The purpose of this service is to provide a basic point-to-point path creation service between two ports specified by TNA addresses.

Input parameters

Path	
Description	The parameters required for creating the path
XML Type	PathType
Java Type	PathType
Multiplicity	Once per request
Mandatory	Yes
Details	Source and destination TNA, bandwidth

Output values

PathIdentifier	
Description	An ID which identifies the created path in later requests
XML Type	PathIdentifierType
Java Type	PathIdentifierType
Multiplicity	Once per request
Mandatory	Yes
Details	Cf. description of PathIdentifierType

Exceptions

Unexpected Fault	Unexpected Error
sourceTNAFault	Source TNA does not exist or is in use or interface type does not match interface
destinationTNAFault	Destination TNA does not exist or is in use
createpathFault	Path could not be created, e.g. bandwidth not available
bandwidthFault	Required bandwidth not supported by endpoint

1.2.2.3 *Termination service (terminatePath)*

This service tears down a point-to-point path which has been set up by an NRPS.

Input Parameters

LSPhandle	
Description	The ID referring to the path which should be terminated
XML Type	PathIdentifierType
Java Type	PathIdentifierType
Multiplicity	Once per request
Mandatory	Yes
Details	Cf. description of PathIdentifierType

Output values

This service does not return any data. A fault is indicated by an exception.

Exceptions

Unexpected Fault	Unexpected Error
pathNotFoundFault	Path does not exist

1.2.2.4 *Path monitoring service (getPathStatus)*

This service provides status information about the specified path.

Input parameters

LSP handle	
Description	The ID referring to the path which should be checked
XML Type	PathIdentifierType
Java Type	PathIdentifierType
Multiplicity	Once per request
Mandatory	Yes
Details	Cf. description of PathIdentifierType

Output values

Path	
Description	The parameters defining the path
XML Type	PathType
Java Type	PathType
Multiplicity	Once per request
Mandatory	Yes
Details	Cf definition of PathType

Exceptions

Unexpected Fault	Unexpected Error
pathNotFoundFault	Path does not exist. Either it does not exist any longer (network error) or it has not been previously created.

1.2.2.5 *Path discovery service (getPathDiscovery)*

This service retrieves any established point-to-point connections in the controlled network. It is basically needed to refresh the NRPS view on the network in case of memory losses through system failures or maintenance reboots.

Input parameters

The service request is sent without any parameters.

Output values

Path List	
Description	A list of all paths in the network is returned
XML Type	PathIdentifierType
Java Type	PathIdentifierType
Multiplicity	Zero or multiple times per request
Mandatory	Yes
Details	A path is defined by its path identifier. If there are no paths an empty list is returned.

Exceptions

Unexpected Fault	Unexpected Error

1.2.2.6 *Endpoint discovery service (getEndpointDiscovery)*

This service retrieves information about any endpoints in the controlled network. It is needed to deliver the available endpoints to the NRPS during system initialization and to refresh the NRPS view on the available endpoints in case of memory losses through system failures or maintenance reboots.

Input Parameters

The service request is sent without any parameters.

Output values

Endpoint List	
Description	Information about endpoints
XML Type	EndpointType
Java Type	EndpointType
Multiplicity	Zero or multiple times per request
Mandatory	Yes
Details	Cf description of EndpointType. If there are no endpoints an empty list is returned.

Exceptions

Unexpected Fault	Unexpected Error

1.2.2.7 *Registration service for receiving update messages*

This service registers the NRPS to receive messages from the web service in case of path status changes or topology changes.

Input parameters

Destination	
Description	The destination address for the push messages
XML Type	String
Java Type	String
Multiplicity	Once per request
Mandatory	Yes
Details	The string contains the URL of the destination to which the push messages are sent

Output values

This service does not return any data. A fault is indicated by an exception.

Exceptions

Unexpected Fault	Unexpected Error
registrationUnsuccessfulFault	Path does not exist

1.2.3 Notifications

1.2.3.1 *List of notifications*

Notification Services	
Path delete notification	The event driven path delete service informs all registered systems, if a path is no longer available.
Endpoint update notification	The event driven endpoint update service provides all registered systems when endpoints have been removed from or added to the controlled network domain.

1.2.3.2 *Path delete notification (pathDelete)*

The event driven path delete notification service informs all registered and authenticated systems, when a path is no longer available.

Path delete message

PathIdentifier	
Description	Identifier of the path which is no longer available
XML Type	PathIdentifierType
Java Type	PathIdentifierType
Multiplicity	Once per message
Mandatory	Yes
Details	Cf. description of PathIdentifierType

1.2.3.3 *Endpoint update notification (endpointUdate)*

The event driven endpoint update notification service provides information to all registered and authenticated systems when endpoints have been removed from or added to the controlled network domain.

Endpoint update message

EndpointID	
Description	Endpoint to be added or deleted
XML Type	EndpointType
Java Type	EndpointType
Multiplicity	Once per message
Mandatory	Yes
Details	Cf. description of EndpointType

Action	
Description	Indicates, which action is associated with the endpoint
XML Type	Int
Java Type	Int
Multiplicity	Once per message
Mandatory	Yes
Details	Actions are specified by integers: 1: endpoint has been removed or is no longer available 2: new endpoint

1.3 GMPLS driver database

All Information about paths, endpoints and devices is kept in a MySQL database, which is only accessed by the core component of the GMPLS driver. It can be administered through any database frontend like phpMyAdmin. The tables have the following structure:

The endpoint table contains all the information related to the endpoints. It is linked to the device table, which contains additional information about the network device to which the endpoint belongs. The path table contains the currently active paths and is linked to the endpoint table, which contains additional information about the endpoints of the path.

1.3.1   Endpoint table

The endpoint table contains fields which are exported via the Endpoint discovery service to the upper layers, e.g. the NRPS or NSP:

Table endpoint – general fields	
endpointID	TNA of the endpoint (unique)
name	Any string, the symbolic name of the endpoint (unique)
description	Empty field or endpoint description, i.e. location, port, etc.
interface	Type of Interface (User or Border)
domain	The domain identifier of the endpoint
bandwidth	The maximal bandwidth supported by the endpoint

Besides the general fields some internal fields are defined, which are used to provide more information to the GMPLS modules to enable them to create and delete paths.

Table endpoint – specific fields	
deviceName	A symbolic name of the network device to which the endpoint belongs (unique)
module	The name of the GMPLS module providing the communication with the GMPLS control plane
interfaceType	The technology of the physical interface to which the endpoint belongs, i.e. GE, 10GE, STM16c
databearer	The identifier of the endpoint physical interface

The endpoint table can be filled by a database frontend application or some of the fields may also be updated from information retrieved by the GMPLS modules. Currently none of the GMPLS modules delivers information about the endpoints.

Project: PHOSPHORUS
Deliverable Number: D.1.3
Date of Issue: 14/11/07
EC Contract No.:   034115
Document Code: Phosphorus-WP1-D.1.3

20

1.3.2   Path table

The path table contains information about all active paths within the GMPLS domains controlled by the GMPLS driver. Besides the information provided when invoking the path creation service the table contains some vendor specific information provided by the GMPLS modules.

Table path– general fields	
srcTNA	The TNA of the source of the path
destTNA	The TNA of the destination of the path
bandwidth	The bandwidth of the path

Besides the general fields some internal fields are defined, which are use to provide more information to the GMPLS modules to enable them to manage paths.

Table path – specific fields	
pathId	A number identifying the path (unique)
vPathIndex	A GMPLS module specific path index, e.g. the control plane path index
pathDescriptor	A label describing the path, e.g. a string built from source and destination TNAs

The path table is updated by the path creation and path termination services. In a later version of the GMPLS driver the path table could be rebuilt in case of failures from the GMPLS modules. Currently none of the GMPLS modules delivers information about existing paths.

1.3.3   Device table

The device table contains information about the devices containing the endpoints. The information contains the IP-address for accessing the device and some vendor specific information about model and software releases.

Table device– general fields	
deviceName	A symbolic name of the device (unique)
ipAddress	The IP-address of the device. It is used for management operations and for retrieving information from the device.
DeviceType	The role of the device, e.g. cross-connect (CC) or UNI-Client (UNI-C)
vendorName	The name of the vendor of the device
model	A vendor specific string, e.g. model number of the device
softwareRelease	The version of the software running on the device

The device table can be filled by a database frontend application or some of the fields may also be updated from information retrieved by the GMPLS modules. Currently none of the GMPLS modules delivers information about the devices.

1.4 GMPLS driver vendor modules

1.4.1 Alcatel 1678 GMPLS SPC module

The Alcatel 1678 GMPLS SPC module allows the creation and deletion of Ethernet Soft Permanent Connections (SPC). The SPCs are configured by using the Alcatel GMRE interface [GMRE], a command line interface, which is accessible via telnet. Via the GMRE, SPCs can be configured locally on a cross connect or spanning multiple cross connects. Signalling is based on a pre-release of the UNI 2.0 standard. The OSPF-TE and RSVP-TE protocols are used for signalling. Signalling may span multiple domains by using E-NNI. Each Ethernet path consists of an Ethernet control connection and one to seven data connections for VC4 containers (cf. Figure 3). The number of the data connections depends on the required bandwidth, i.e. as one VC4 container can carry 155 Mbps, the actual bandwidth is a multiple of 155 Mbps. The user can specify every bandwidth up to 1 Gbps. The system automatically chooses the number of required VC4 containers for transmitting the requested bandwidth.
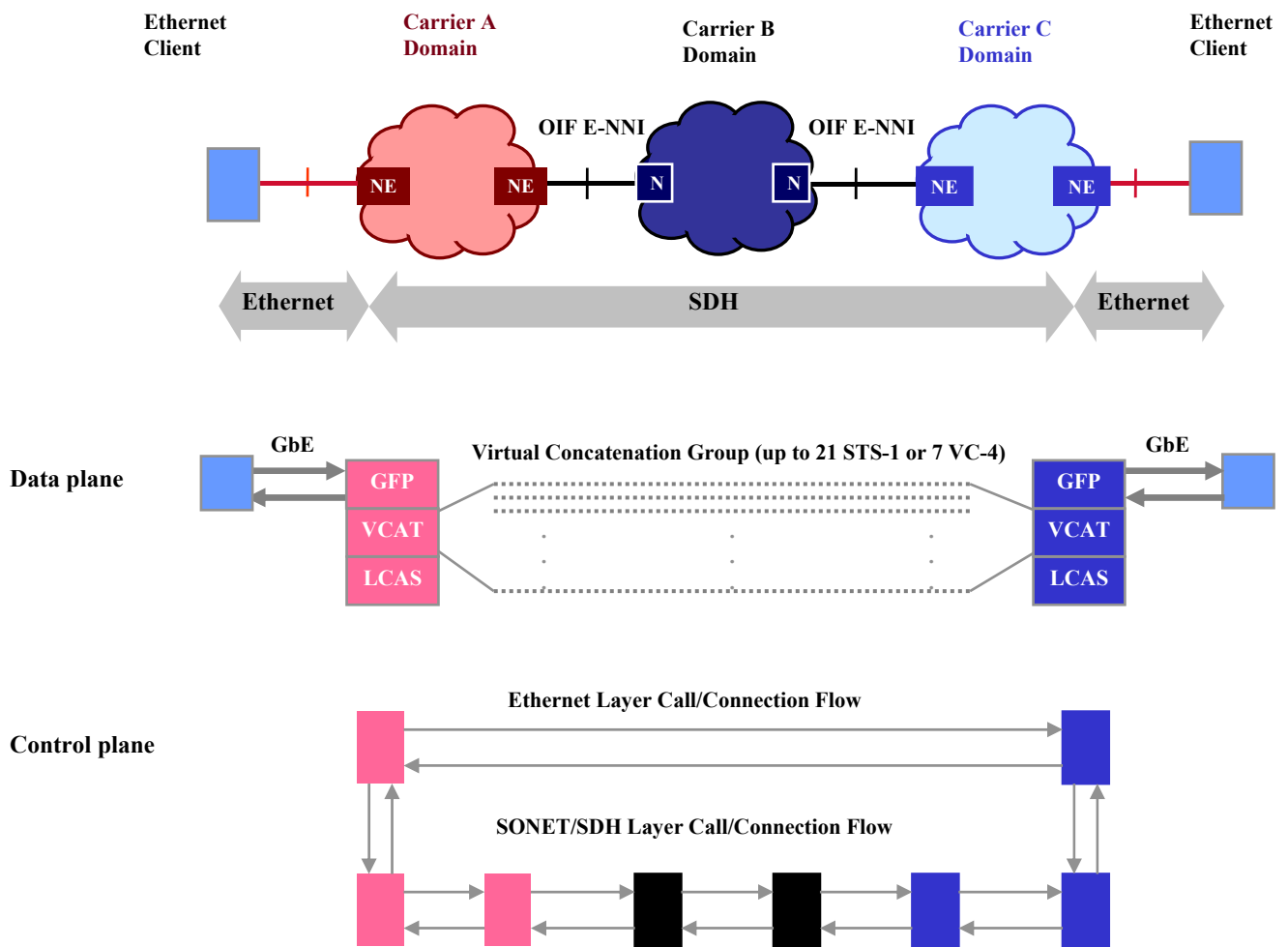
Figure 3: Ethernet Soft Permanent Connection [TNC-2007]

An SPC is created and deleted by configuring the network element given by the source TNA. At this network element the following commands are entered in order to create and delete an SPC:

Example of a create command:

config lsp lspdescriptor "-sa-0408-bn-0301" lsptraffic GBETHERNET inlink 13040801 inlabel 7-0-0-0-0 prot_rest_type unPROTECTED sourceTNA 10.3.3.8 destTNA 10.3.1.1 adminState up

Example of a delete command:

config lsp lspindex 19200 adminState down
delete lsp 19200

Currently the Alcatel Ethernet SPC implementation does not support Explicit Route Objects (ERO). EROs are used to specify explicit routes through a GMPLS domain. The NRPS needs these explicit routes for creating pre-computed paths in case of advance reservation. As EROs are not supported, the Alcatel SPC module does not return any information about links.

1.4.2   Alcatel proxy UNI 2.0 module

The Alcatel 1678 GMPLS SC module allows the creation and deletion of Ethernet Switched Connections (SC). The SCs are configured by using the Alcatel Proxy UNI 2.0 clients [UNI], which are configured via an SNMP interface. Via the UNI clients, SCs between the network devices to which the UNI clients are connected can be configured. Signalling is based on a pre-release of the UNI 2.0 standard. The OSPF-TE and RSVP-TE protocols are used for signalling. Signalling may span multiple domains by using E-NNI. Each Ethernet path consists of an Ethernet control connection between the UNI clients and one to seven data connections for VC4 containers between the network elements hosting the UNI-N (cf. Figure 4). The number of the data connections depends on the required bandwidth, i.e. as one VC4 container can carry 155 Mbps, the actual bandwidth is a multiple of 155 Mbps. The user can specify every bandwidth up to 1 Gbps. The system automatically chooses the number of required VC4 containers for transmitting the requested bandwidth.

An SC is created and deleted by configuring the UNI client given by the source TNA. The following SNMP commands are sent to the UNI client in order to create and delete an SC:

Example of a create command:

snmpset -c private -v 2c -ls 10 -lS tmnxGmplsTunnel 192.168.20.18 RowStatus = 4 SourceTNA = 10.3.3.4
DestinationTNA = 10.3.1.4 HoBandwidth = undefined SwitchingType = 51 LSPEncoding = 2
ResourcePointer = 7

Example of a delete command:

snmpset -c private -v 2c -ls 10 -lS tmnxGmplsTunnel 192.168.20.18 RowStatus = 6

The Management Information Bases [MIB] used by SNMP for creating and deleting SCs are proprietary MIBs from Alcatel, which are based on the general GMPLS MIBs.
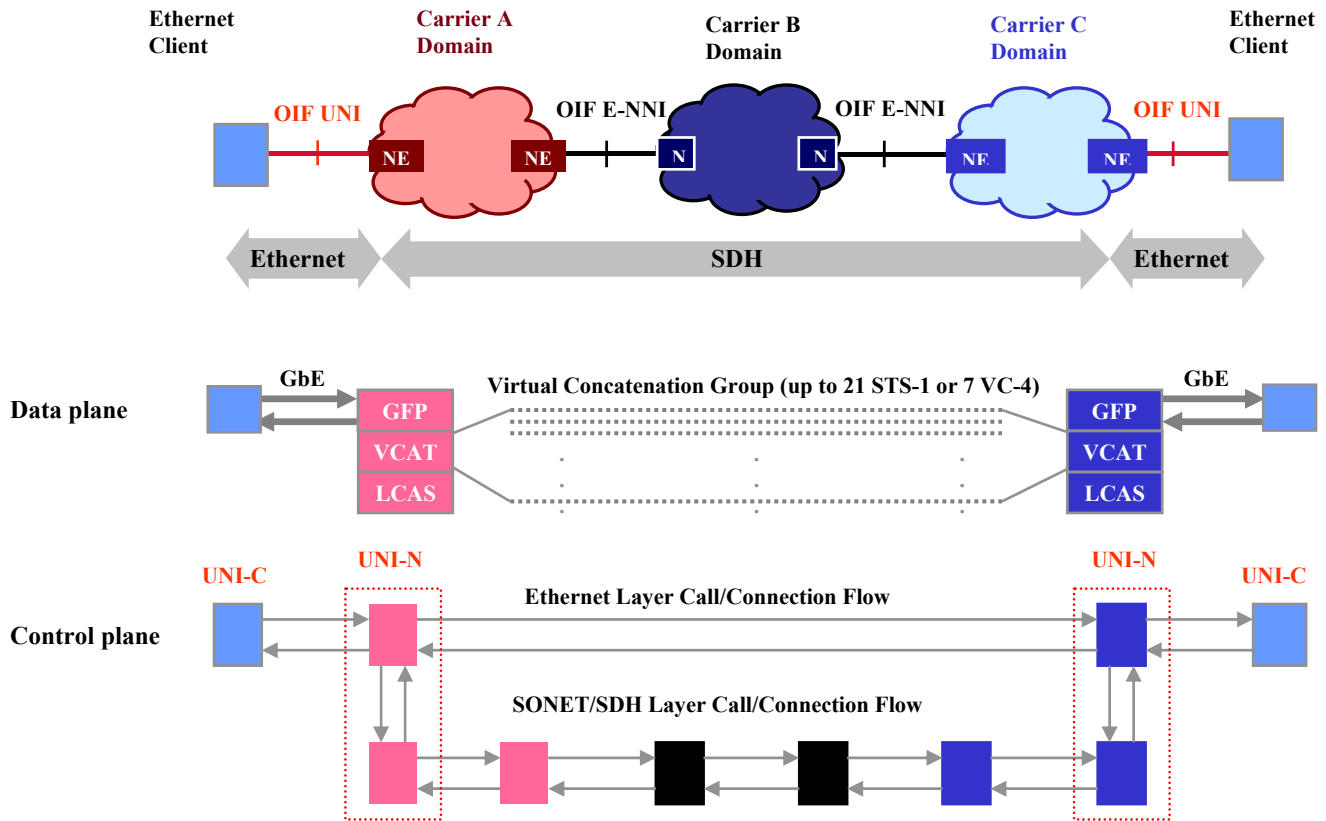
Figure 4: Ethernet Switched Connection [TNC-2007]

1.4.3   Dummy module

The purpose of the dummy module is to simulate a GMPLS domain. Requests for connections are always granted. Termination of requests will always be successful, too. The dummy module may be used for testing the GMPLS driver with new domains, where only the endpoints are known. No data is transmitted at the data plane.

1.5   GMPLS driver Java client

The GMPLS driver Java client is a web service client, which can be used to test the GMPLS driver web service. It allows creating and deleting paths via a graphical user interface. During path creation a list of all endpoints is

presented to the user. A list of active paths is shown in the show paths menu, from which paths can also be deleted. Figure 5 shows the path creation interface and Figure 6 shows the path show/delete interface.
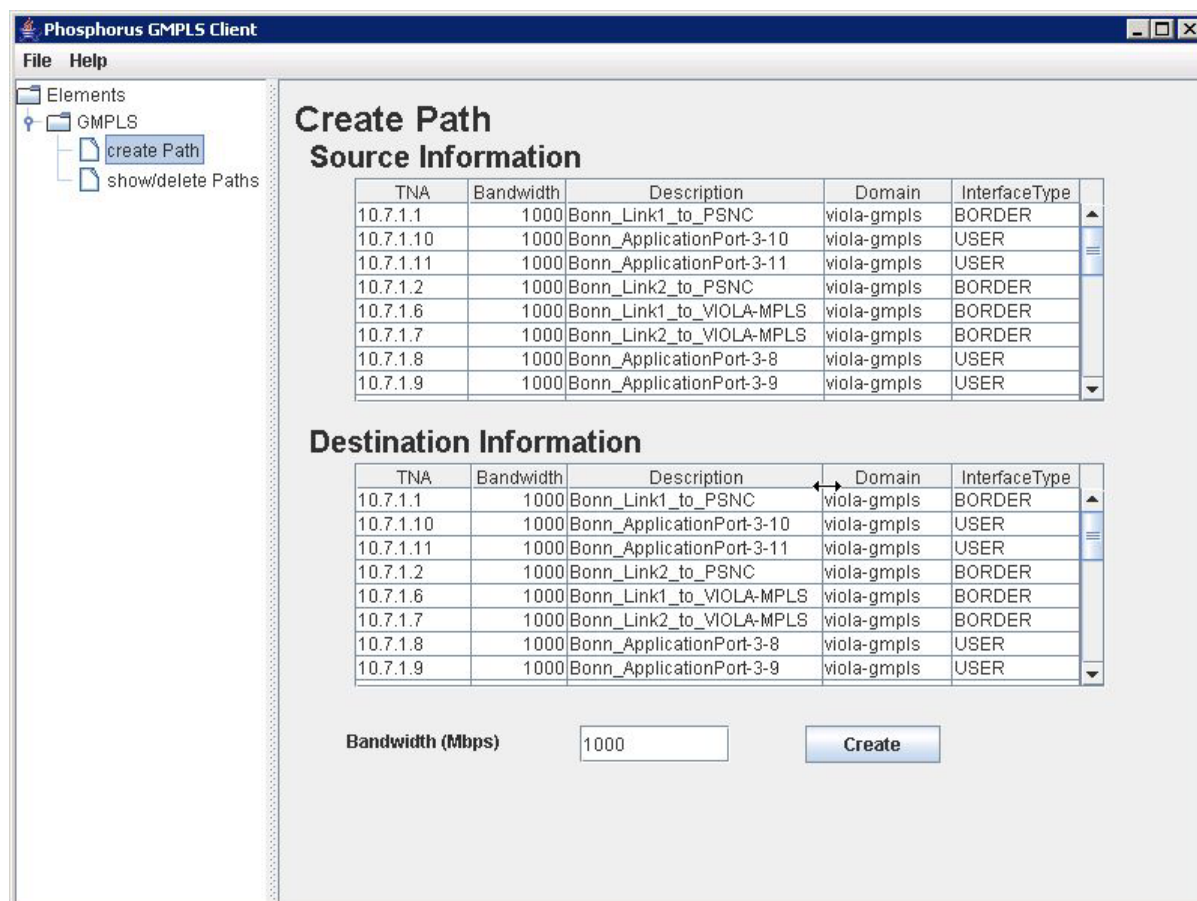

Figure 5: GMPLS driver client create path menu
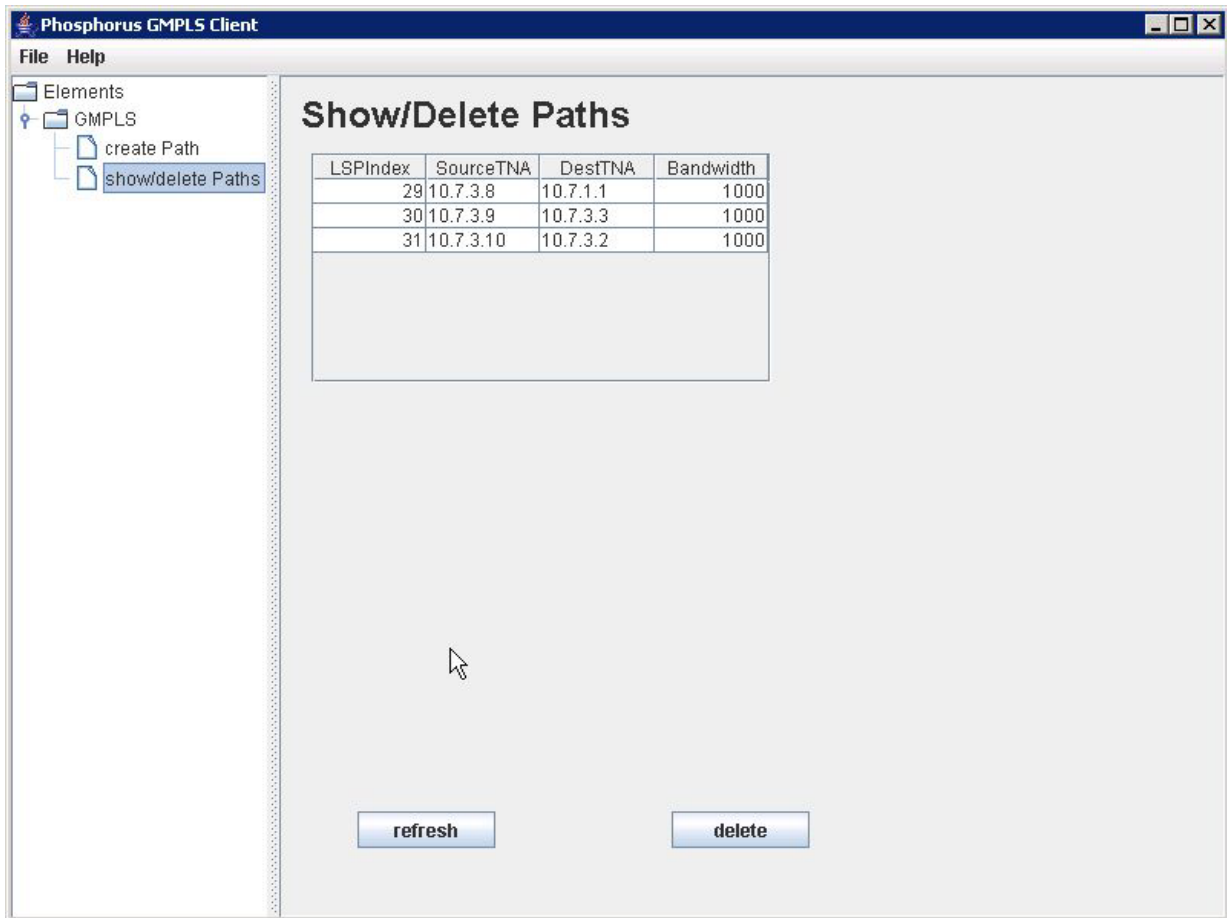
Figure 6: GMPLS driver client show/delete paths menu

2   Thin NRPS

The Thin NRPS is a network resources provisioning system for domains with a GMPLS control plane. Its relationship to the other PHOSPHORUS modules is shown in Figure 7. The Thin NRPS,

-    Provides a reservation web service, which is used by the NSP to reserve, create and delete network connections. It uses the same web service interface as the NRPS adapters.

-    Provides a notification receiver interface, which enables the GMPLS driver to send update information about endpoints and paths.

-    Acts as a client to the topology manager web service of the NSP, e.g. it will add a domain and provide information about endpoints.

-    Acts as a client to the GMPLS driver web service for creating and deleting network connections and for obtaining information about endpoints and existing connections.

Internally, the Thin NRPS provides the following functions:

-    Handles reservation requests from the NSP; checks, if the reservation can be granted and keeps track of the reservations in a database.

     As the Thin NRPS currently only gets endpoint information from the underlying GMPLS driver, it has no knowledge about internal links and their usage. This means, that in case of advance reservation no checks for availability of bandwidth on the internal links can be performed. Therefore the Thin NRPS will only check, if endpoints are available and if there are conflicts concerning the usage of endpoints within overlapping reservations.
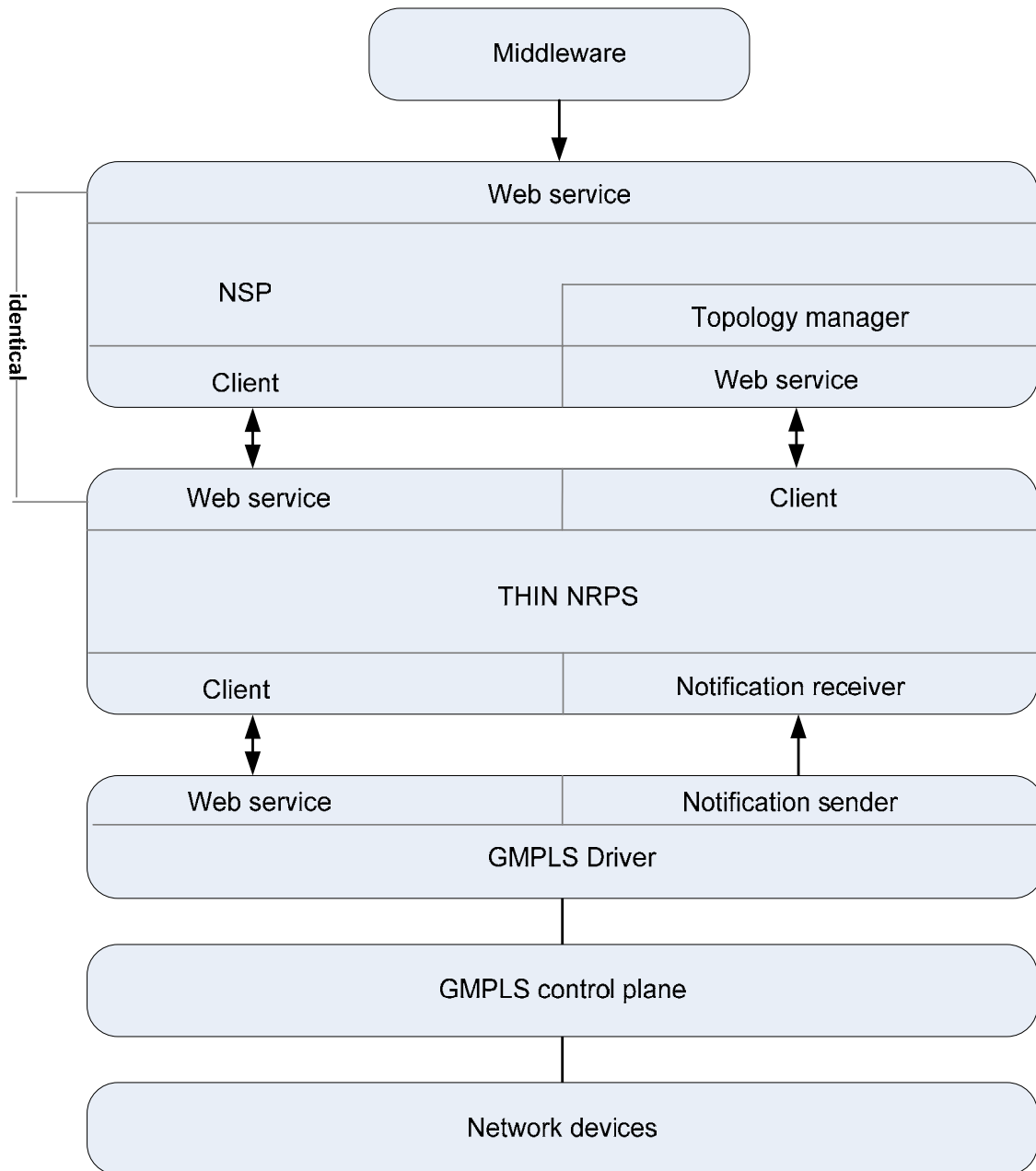
-    Schedules creation and termination of network connections using the GMPLS driver.

-    Registers its domain at the NSP domain manager, retrieves user and border endpoints from the GMPLS driver and forwards border endpoints to the NSP domain manager.

- Handles endpoint updates from the GMPLS driver and informs the NSP domain manager.

Figure 7: Thin NRPS and other PHOSPHORUS modules

In contrast to a normal NRPS the Thin NRPS has only a limited functionality. Only fixed reservations are currently supported, because no topology information about links and their usage is available. Therefore the following functions are currently not supported:

- Deferrable reservations

- Malleable reservations

- Unidirectional or point-to-multipoint connections

- MaxDelay

- DataAmount.

If one of these functions is specified, an exception is thrown.

Within the following chapters the various components of the Thin NRPS are described in more detail.

2.1 Thin NRPS reservation web service

The Thin NRPS Reservation web service is identical to the NRPS adapter reservation web service, which is identical to the NSP reservation web service. For more information see the description of the NSP reservation web service.

2.2 Thin NRPS notification receiver

The Thin NRPS notification receiver handles notifications, sent by the GMPLS driver. These notifications include updates on endpoints (add, delete) and notifications about prematurely terminated connections.

2.3 Thin NRPS NSP topology manager client

The Thin NRPS topology manager client provides domain information to the NSP using the NSP topology manager services. The required domain and topology information is obtained from the GMPLS driver, which delivers border and user endpoints. Only the border endpoints are propagated to the NSP topology manager. If endpoints are added or deleted within the GMPLS driver, the Thin NRPS topology update web service receives the updated information and passes it to the NSP topology manager.

The domain information is obtained from the domainId field of an endpoint definition. The flow of information is explained in Figure 8.

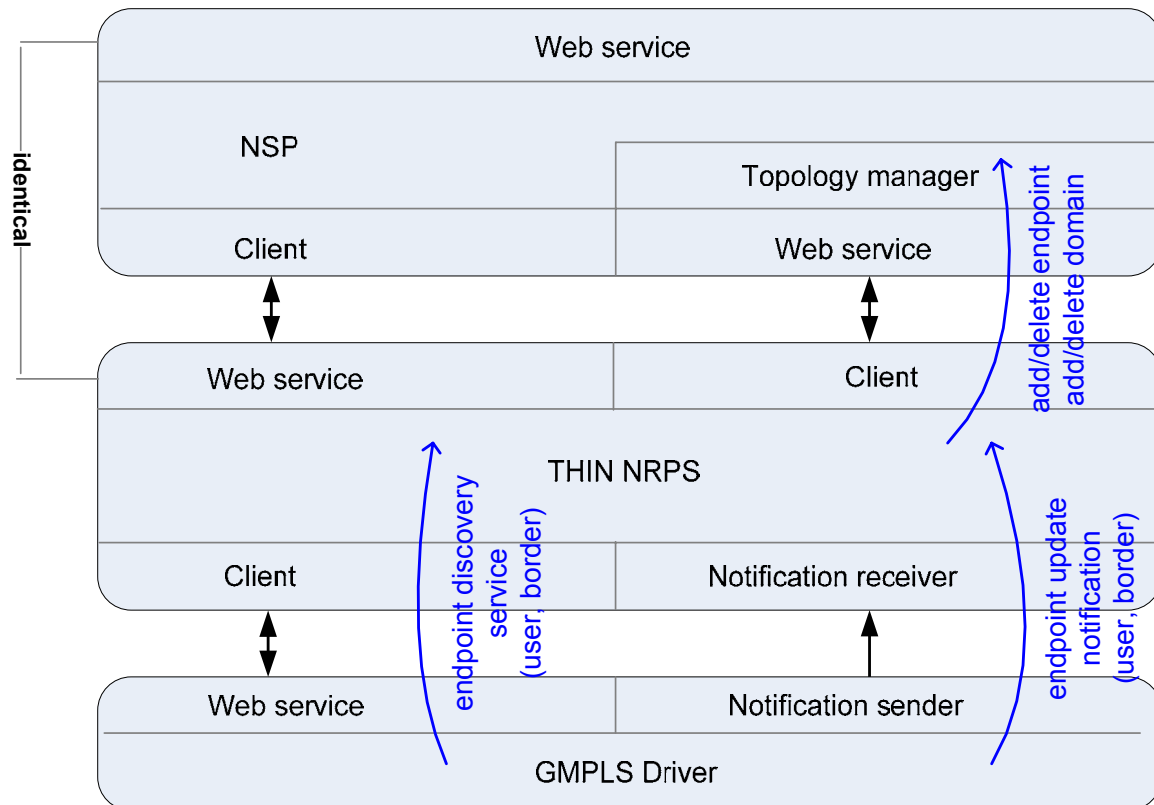|	
Project:	PHOSPHORUS
Deliverable Number:	D.1.3
Date of Issue:	14/11/07
EC Contract No.:	034115
Document Code:	Phosphorus-WP1-D.1.3

30

Figure 8: Distribution of endpoint information

2.4   Thin NRPS GMPLS driver client

The Thin NRPS GMPLS driver client sends path creation and deletion requests to the GMPLS driver. Additionally it retrieves endpoint and path information from the GMPLS driver.
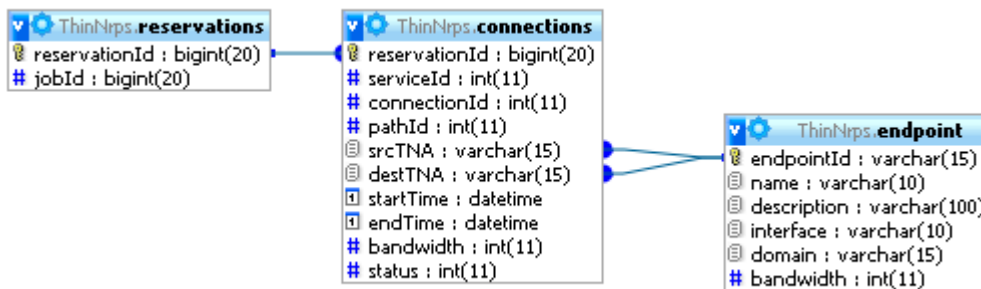
2.5   Thin NRPS database

The Thin NRPS database contains information about the active and reserved network connections. Additionally it has information about the user and border endpoints of a domain. The following tables are used:

Reservations table: Information about jobs and their reservations

Connections table: Information about active or reserved network connections

Endpoint table: Information about endpoints.

2.5.1   Reservations table

The reservations table contains information about jobs and their reservations:

Table reservations	
reservationId	The reservation identifier generated by the Thin NRPS. The reservationIds are local to a job. The connections belonging to a reservation are kept in the connections table.
jobID	The job identifier specified by the user. A job may have several reservations.

2.5.2   Connections table

The connections table contains information about the current and future network connections. The following fields are defined:

Table connections	
reservationId	The reservation identifier. A reservation may contain several services.
serviceId	The service identifier. A service may contain several connections.
connectionId	The connection identifier
pathId	The path identifier. A GMPLS path corresponds to a connection. The path identifier is returned by the GMPLS driver after successful creation of a path.

Project:	PHOSPHORUS
Deliverable Number:	D.1.3
Date of Issue:	14/11/07
EC Contract No.:	034115
Document Code:	Phosphorus-WP1-D.1.3

32

srcTNA	The TNA of the source of the connection
destTNA	The TNA of the destination of the connection
startTime	The time, when a connection starts
endTime	The time, when a connection ends
bandwidth	The bandwidth of the path
status	The status of a connection specified as a bit vector. Values are: 0: the connection has not yet been scheduled 1: the connection is scheduled 2: connection is autoactivated 4: connection is active.

2.5.3 Endpoint table

The endpoint table contains fields, which are retrieved via the Endpoint discovery service from the GMPLS driver at the initialization process of the Thin NRPS. During the lifetime of the Thin NRPS endpoints can be removed and inserted via the notification receiver. Information about border endpoints is sent to the web service of the NSP topology manager at start up and whenever new endpoints are inserted or existing endpoints are deleted.

Table endpoint	
endpointID	TNA of the endpoint (unique)
name	Any string, a symbolic name of the endpoint (unique)
description	Empty string or endpoint description, i.e. location, port, etc.
interface	Type of Interface (user or border)
domain	The domain identifier of the endpoint
bandwidth	The maximal bandwidth supported by the endpoint

2.6 Thin NRPS reservation handler

The Thin NRPS reservation handler processes all reservation requests initiated by the NSP. It is based on the ARGON NRPS adapter reservation handler.

Project: PHOSPHORUS
Deliverable Number: D.1.3
Date of Issue: 14/11/07
EC Contract No.:   034115
Document Code: Phosphorus-WP1-D.1.3

33

2.7 **Thin NRPS connection handler**

The Thin NRPS connection handler handles single connections. It provides the following functions:

- Checks if connections are possible, by checking the availability of endpoints within a given interval

- Sends creation and termination requests for connections to a scheduler

- Maintains the connections table.

3 Test environment

The GMPLS driver and the Thin NRPS have been tested by using the JUnit testing framework. The GMPLS has been tested by using the Java client, too.

The tests have been performed within the GMPLS domain of the VIOLA PHOSPHORUS test-bed, which is shown in Figure 9. Alcatel 1678 cross connects are controlled by the Alcatel GMPLS implementation GMRE. Soft Permanent Connections (SPC) between the Gigabit Ethernet endpoints were created, monitored and terminated via the Thin NRPS and the GMPLS driver.
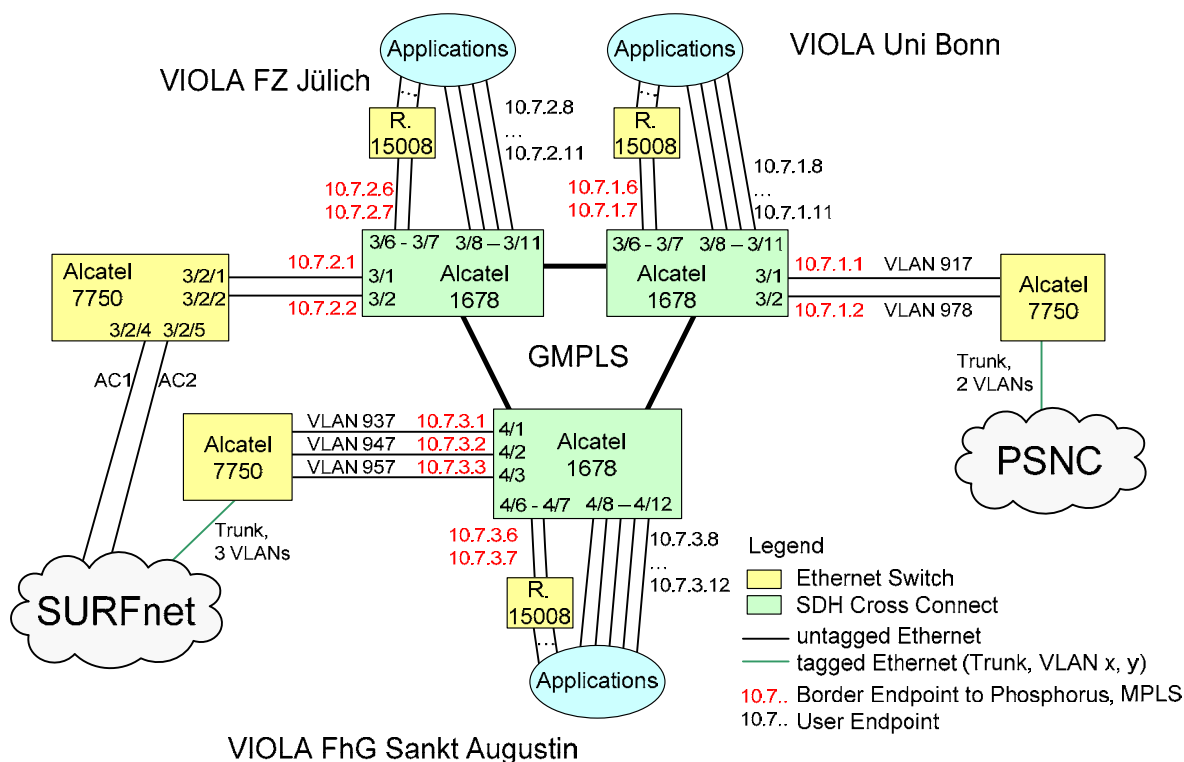

Figure 9: GMPLS test environment of the VIOLA PHOSPHORUS test-bed

4   GMPLS and NRPS

Network Resources Provisioning Systems (NRPS) like ARGON, DRAC and UCLP normally have complete control over the network managed by them. At each time they have a complete overview of the topology of the network. They know, which network elements (NE), ports and links are available and they know the current status, e.g. the current link utilization. Normally they provide point-to-point connections between two endpoints (user ports) for a predefined time including immediate or advance reservation. Recovery connections can be provided, too. Normally, NRPSs configure NEs directly for creating network connections. This is done by using Command Line Interfaces (CLI) or by using the Simple Network Management Protocol (SNMP). Normally the NEs do not exchange any control information among themselves. Control or management information is only exchanged between the NRPSs and the NEs.

In contrast to this centralized approach, networks with a control plane operate in a decentralized fashion. In the case of the ASON/GMPLS/OIF based control plane the OSPF-TE protocol is used for distributing node, link, link utilization and endpoint (TNA) information. Network connections are created by using the RSVP-TE signalling protocol. In case of network failures connections can be rerouted by signalling an alternate path.

If we compare NRPS and GMPLS control plane based networks we see that they provide nearly the same functionality. The difference is that NRPS based systems use a centralized approach, whereas GMPLS based systems use a decentralized approach. Currently GMPLS based systems do not support advance reservation. This missing feature will be developed within the G²MPLS development of the PHOSPHORUS WP2.

Are NRPSs for GMPLS based networks meaningful as they provide the same functionality? Yes, currently this combination is useful if advance reservation is required. But both approaches can only coexist, if the path calculation is only done by the NRPSs. This means, that the GMPLS control plane must provide the following services to the NRPSs:

-       Provision of topology information, i.e. information about NEs, links and endpoints. There must also be a feedback function from the GMPLS control plane in case of link or node failures.

-       Availability to create explicit paths using Explicit Route Objects (ERO) in the GMPLS network. Explicit paths are pre-calculated paths, which have been calculated by the NRPSs and provided as EROs to the GMPLS control plane.

Currently the GMPLS driver only delivers topology information about endpoints, because there is currently no GMPLS control plane within PHOSPHORUS which supports EROs. But the GMPLS driver can be extended to provide additional topology information, if this functionality is necessary. In deliverable D1.1 the required data types and web services have been described.

5 References

[ARGON]	- ARGON – Allocation and Reservation in Grid-enabled Optic Networks" http://www.viola-testbed.de/content/index.php?id=reports
	- Project home page: http://www.viola-testbed.de
	- B2.3.2 (2006) "Program Documentation of the Reservation System User Interface" http://www.viola-testbed.de/content/index.php?id=reports
[DRAC]	http://www.nortel.com/solutions/optical/collateral/nn110181.pdf
[GN2-JRA3]	"Deliverable DJ.3.3.1:GÉANT2 Bandwidth on Demand Framework and General Architecture" http://www.geant2.net/upload/pdf/GN2-05-208v7_DJ3-3-1_GEANT2_Initial_Bandwidth_on_Demand_Framework_and_Architecture.pdf
[GT4]	http://www.globus.org/toolkit/
[UCLPv2]	Project home page: http://www.uclp.ca/
[GMRE]	Alcatel 1678MCC, Rel. 3.1B, GMPLS/GMRE, Command Line Interface User Guide
[UNI]	Alcatel Manual for ProxyUniC 2.0
[UNICORE]	http://www.unicore.eu
[TNC 2007]	On-Demand Provisioning of Ethernet Services over Multiple Network Domains by Using OIF UNI/E-NNI Inter-Domain Interfaces: http://tnc2007.terena.org/programme/presentations/show.php?pres_id=73

Project: PHOSPHORUS
Deliverable Number: D.1.3
Date of Issue: 14/11/07
EC Contract No.:   034115
Document Code: Phosphorus-WP1-D.1.3

38

6 Acronyms

[AAA]	Authorization, Authentication and Accounting
[ARGON]	Allocation and Reservation in Grid-enabled Optic Networks
[ASON]	Automatically Switched Optical Network
[BoD]	Bandwidth on Demand
[CP]	Control Plane
[DB]	Database
[DM]	Domain Manager
[DRAC]	Dynamic Resource Allocation Controller
[E-NNI]	Exterior NNI
[ERO]	Explicit Route Objects
[GLIF]	Global Lambda Integrated Facility
[GMPLS]	Generalized Multi Protocol Label Switching
[GMRE]	Name of Alcatel's GMPLS implementation
[IDM]	InterDomain Manager
[ID]	Identifier
[IP]	Internet Protocol
[I-NNI]	Interior NNI
[JRA]	Joint Research Activity
[LSA]	Link State Attribute
[LSP]	Label Switched Path
[MAC]	Media Access Control
[MIB]	Management Information Base
[MS]	MetaScheduler
[NNI]	Network-Network Interface
[NRPS]	Network Resource Provisioning System
[NSP]	Network Service Plane
[NSAP]	Network Service Access Point
[OIF]	Optical Internetworking Forum
[OSPF]	Open Shortest Path First
[QoS]	Quality of Service
[RSVP]	ReSerVation Protocol

[SC]	Switched Connections
[SDH]	Synchronous Digital Hierarchy
[SPC]	Soft Permanent Connections
[TE]	Traffic Engineering
[TXN]	Transaction
[TNA]	Transport Network Address
[UCLPv2]	User Controlled LightPaths version 2
[UNI]	User-Network-Interface
[URL]	Uniform Resource Locator
[VLAN]	Virtual Local Area Network
[VPLS]	Virtual Private LAN Services
[WP]	Work Package
[WS]	Web Service

7   Appendix

7.1   Terminology

In these paragraphs different layers used in the architecture will be defined, as well as the most important concepts that will be used in this document:

- Grid Layer: This layer will be developed within WP3 and will be based on the Globus Toolkit 4 [GT4] and UNICORE [UniCore] software. It is responsible for running a set of applications defined by WP3, it also forwards to the Network Service Plane the network requirements to run these applications. The Grid Layer is also referred in this document as Grid Middleware.

- Network Service Plane (NSP): The network service plane is the adaptation layer between the Grid layer and the NRPS layer. It coordinates the different networks in the NRPS layer to meet network resources requests coming from the Grid Layer. This block is a key element to provide interoperability between the different NRPSs, GMPLS and the Grid Layer.

- NRPS Layer: This layer is composed by several Network Resource Provisioning Systems managing different domains. For the scope of the PHOSPHORUS project these NRPSs can be either an ARGON [ARGON], a DRAC [DRAC] or a UCLPv2 [UCLPv2] system. Each system is responsible for the management of the resources available in its domain. This management can be achieved by directly accessing the nodes of the transport network, or, if available, by means of a GMPLS control plane that configures the network.

- NRPS Adapter: Since each NRPS has a particular interface, an adapter between the NRPSs and the NSP is needed. The NRPS Adapter is responsible for the mapping of these interfaces to a common interface.

- Transport Network (TN): within the PHOSPHORUS project a Transport Network is a group of physical devices and links managed by an entity (an NRPS or a GMPLS CP) that allows the transport of information between endpoints.

- Advance Reservations: All the applications that will be used in WP3 have as a requirement the possibility to do advance reservations of resources. This means that network resources can be requested in advance; therefore an advance reservation includes a Start and End Time indicating when this reservation should take place.