034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable D6.8
New developments testing report

Due date of deliverable: 2008-09-30
Actual submission date: 2008-09-30
Document code: Phosphorus-WP6-D6.8

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
Instytut Chemii Bioorganicznej PAN

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 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)	

Authors

Artur Binczewski	PSNC
Wojbor Bogacki	PSNC
Gino Carrozzo	NXW
Nicola Ciuli	NXW
Yuri Demchenko	UvA
Thomas Eickermann	FZJ
Sergi Figuerola	i2CAT
Marcin Garstka	PSNC
Leon Gommans	UvA
Cees de Laat	UvA
Damian Parniewicz	PSNC
Angel Sanchez	I2CAT
Maciej Stroiński	PSNC
Jan Węglarz	PSNC
Wolfgang Ziegler	FhG

Abstract

Phosphorus addresses some of the key technical challenges to enable on-demand e2e network services across multiple domains. The concept of the Phosphorus network will make applications aware of their complete Grid resources (computational and networking) environment and capabilities, and able to make dynamic, adaptive and optimized use of heterogeneous network infrastructures connecting various high-end resources. Phosphorus will enhance and demonstrate solutions that facilitate vertical and horizontal communication among applications middleware, existing Network Resource Provisioning Systems, and the proposed Grid-GMPLS Control Plane.

One of the main assumptions of Phosphorus is that the project propositions and developments should be validated and demonstrated in a real advanced optical network. To achieve this, the project has built a distributed testbed in which the project outcome will be demonstrated with a set of real scientific applications in a set of real-life scenarios.

This document summarises the results of the validation of the project developments which were done before September 30, 2008. The tests conducted after this date will be reported in next release of this document which will be submitted in month 30 of the project duration..

Table of Contents

List of Figures

List of Tables

Executive Summary

Phosphorus addresses some of the key technical challenges to enable on-demand e2e network services across multiple domains. The Phosphorus network will make applications aware of their complete GRID resources (computational and networking) environment and capabilities, and able to make dynamic, adaptive and optimized use of heterogeneous network infrastructures connecting various high-end resources. Phosphorus will enhance and demonstrate solutions that facilitate vertical and horizontal communication among applications middleware, existing Network Resource Provisioning Systems, and the proposed Grid-GMPLS Control Plane.

One of the main assumptions of Phosphorus is that the project propositions and developments should be validated and demonstrated in a real advanced optical network. To achieve this, the project has built a distributed testbed in which the project outcome is verified with a set of real scientific applications in a set of real-life scenarios. This way the testbed emulates a modern GRID environment in which demanding applications running on computational nodes use a transmission network to exchange data between the nodes and access external devices.

This document summarises the results of the validation of the project developments which were done before September 30, 2008. The tests conducted after this date will be reported in the next release of this document which will be submitted in month 30 of the project duration.

The PHOSPHORUS developments which will be tested in the testbed come from different PHOSPHORUS activities. Tests referring to developments of each workpackage are described in separate chapters. Chapter one describes tests of the HARMONY software developed by Workpackage 1 which integrates the different Network Resource Provisioning Systems existing before the start of PHOSPHORUS (UCLP, DRAC, ARGON). Chapter two describes the first tests of the GRID-GMPLS stack (G^2MPLS) prepared by Workpackage 2 which extends the GMPLS services towards GRIDs. The next chapter describes tests of applications and GRID middleware (developed by Workpackage 3), while chapter four focuses on a test of the security mechanisms (GAA-TK) developed by Workpackage 4 and their integration with developments of workpackage 1.

All the tests described in this document were conducted in the distributed PHOSPHORUS testbed with the use of the resources available in different local testbeds.

Some of the tests were also described in deliverables prepared by the workpackages which submitted their developments for tests. They are described also in this deliverable in order to make the search for information about tests easier – the information can be found in this deliverable as well as in deliverables describing the developments of the other workpackages.

The results of the tests were used by the workpackages which provided their developments for tests in elimination of bugs, planning next phases of developments as well as proving the ideas behind the developmets in a real network.

1 NRPS experiments – WP1

1.1 Tested features of the Harmony system

Since its first description in D1.4, the Network Service Plane implementation has been extended by several features that required testing. In this chapter, the tests of the peer-to-peer topology distribution, peer-to-peer reservation setup and malleable reservations and their corresponding results are described. The following subsections provide a short introduction to the tested features.

1.1.1 Peer-to-peer topology distribution

In the new peer-to-peer operation mode, there is not central instance where all domains register their domain information. Instead, the domain information is flooded to all domains. Each domain registers its information with its peers in regular time intervals. Upon reception of new information from other domains, this information is also forwarded.

The decision whether received domain information is newer than the information in the database is based on a sequence number field that is incremented by the originator of the domain information and on the time when the last domain information was received, the latter in order to recover from the case where an instance is e.g. rebooted and did not store its last sequence number correctly.

1.1.2 Malleable reservations

While fixed reservations with the parameters “starting time” and “duration” are tailored for “interactive” applications, another important application type requires the possibility to make data transfers within a given deadline. For this reservation type, the parameters “earliest starting time”, “deadline”, “data amount” (and “minimum” / “maximum bandwidth”) specify a time window within which the data transfer has to take place. It offers more degrees of freedom, reflecting that it is irrelevant for the application when exactly the transfer takes place and what specific bandwidth is used. This takes the burden of trying to find a time window when the necessary resources are available from the application and places it to the service plane.

1.2 Testbed setup

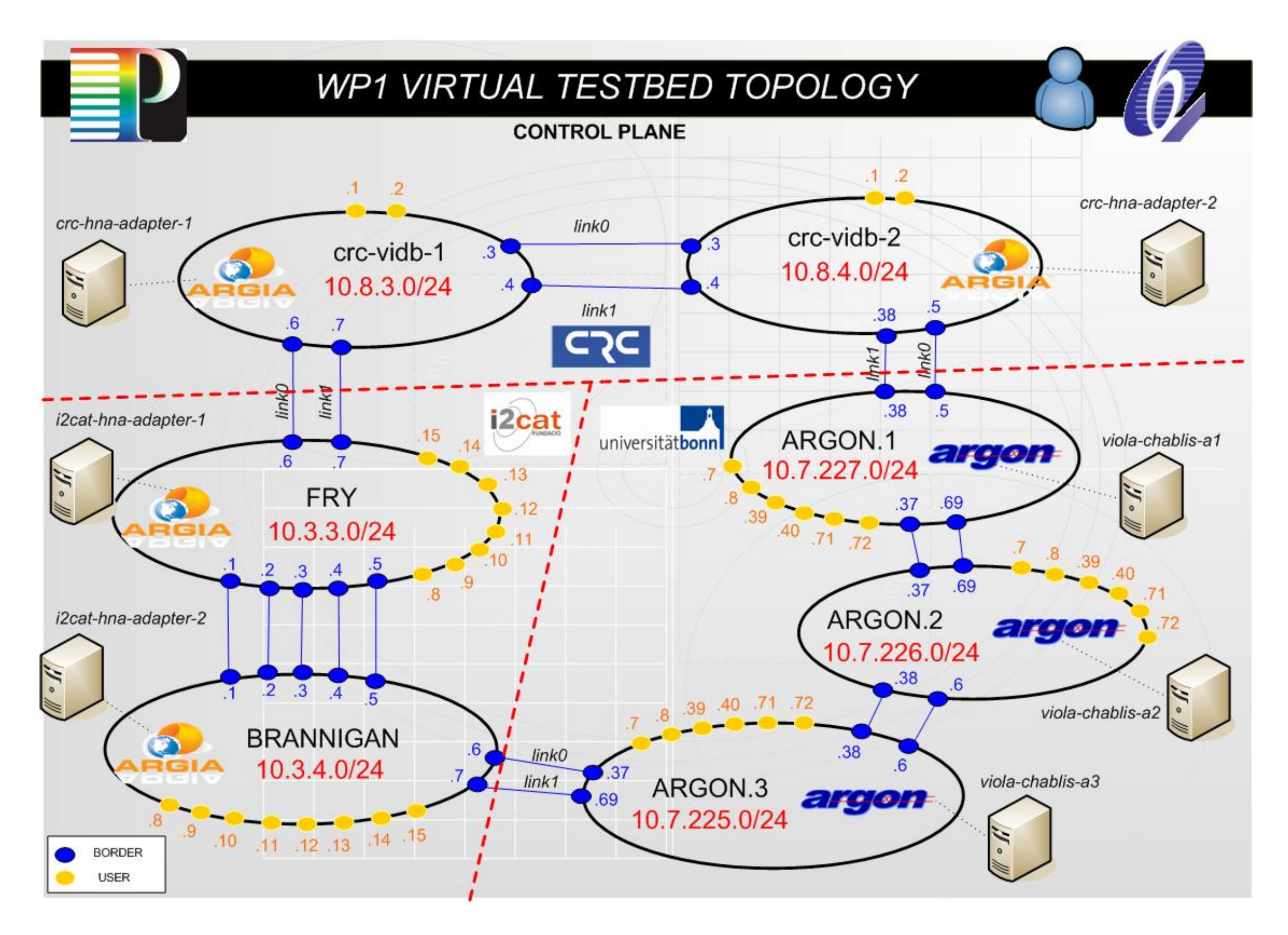

Figure 1.1: WP1 related layers

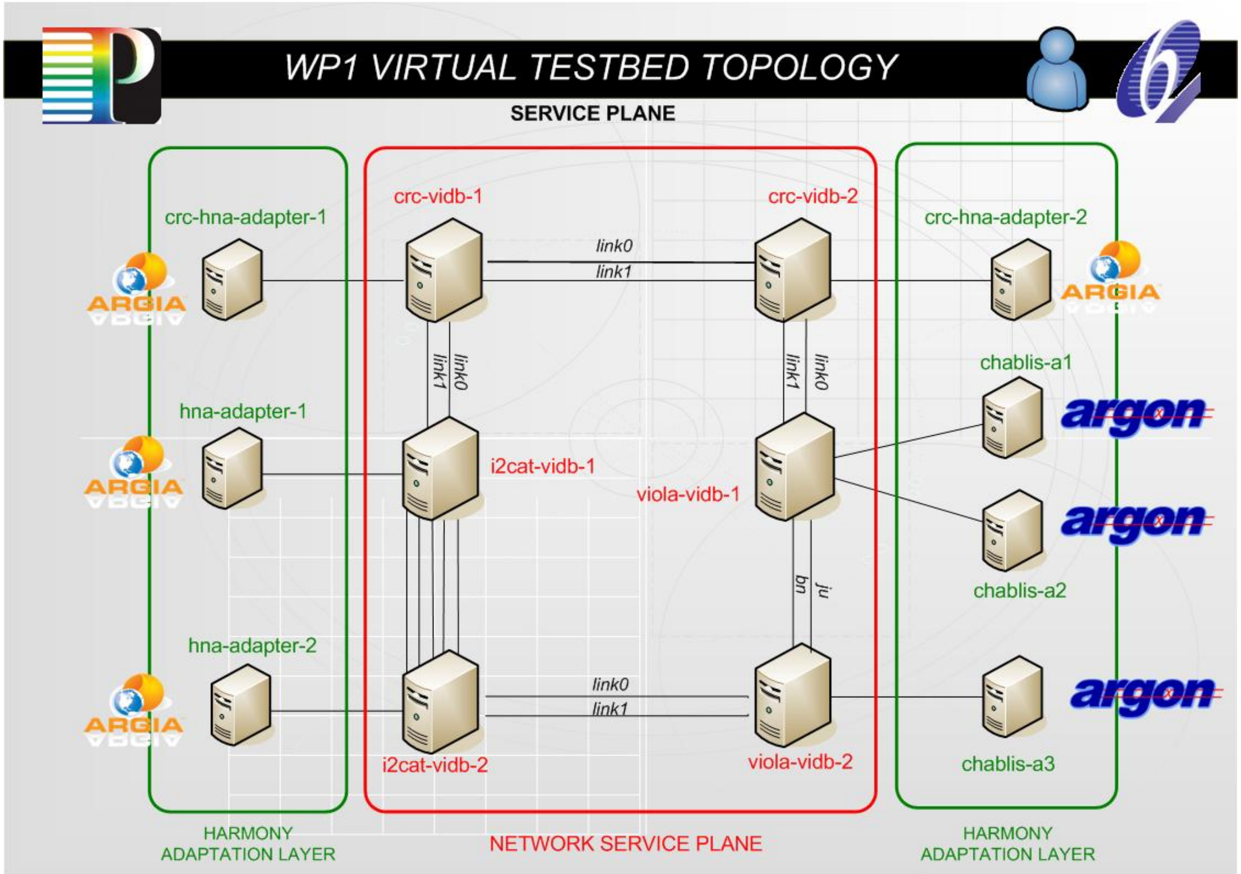

Figure 1.2: WP1 related layers

Figure 1.1 and Figure 1.2 show the virtual testbed setup in which the tests have been carried out. It has been decided not to use the existing testbed with physical links because the tests mainly concern the behaviour of the service plane, so the data plane is not relevant for the tests. From the service plane point of view, there is no difference between the virtual testbed and the real testbed, because communication takes place with the same NRPSs. These NRPSs are merely operated in a test mode in which they do not control real devices, but "dummy" devices.

Each of the partners participating in the virtual testbed (i2Cat, University of Bonn, and CRC) operate two IDBs, each of which is connected to two other IDBs, forming a ring topology. This topology is motivated by the need to have some longer routes, also for testing topology information dissemination. Each IDB controls a single HNA, except for one of the IDBs at the University of Bonn that controls two ARGON subdomains.

Descriptor	Test connection	Blocking connections
i2cat	10.3.3.8 – 10.3.4.8	10.3.3.11 – 10.3.4.11

		10.3.3.12 – 10.3.4.12 10.3.3.13 – 10.3.4.13 10.3.3.14 – 10.3.4.14 10.3.3.15 – 10.3.4.15
viola	10.7.225.7 – 10.7.226.7	10.7.225.71 – 10.7.226.71 10.7.225.72 – 10.7.226.72
crc	10.8.3.1 – 10.8.4.1	10.8.3.2 – 10.7.227.72 10.3.3.15 – 10.8.4.2
i2cat-viola	10.3.4.8 – 10.7.226.7	10.3.4.14 – 10.7.225.71 10.3.4.15 – 10.7.225.72
i2cat-crc	10.3.4.8 – 10.8.3.1	10.3.3.14 – 10.8.4.1 10.3.3.15 – 10.8.3.2
viola-crc (1)	10.7.226.7 – 10.8.3.1	10.7.226.71 – 10.7.227.71 10.7.226.72 – 10.7.227.72
viola-crc (2)	10.7.226.7 – 10.8.3.1	10.7.227.71 – 10.8.4.1 10.7.227.72 – 10.8.4.2

Table 1.1: Test connections and corresponding blocking connections

Table 1.1 shows the test connections that have been used in the following tests, together with additional "blocking connections" that have been setup prior to the actual test connection in some test to force the test connection to be established across the longer path in the cycle of domains.

1.3 Test results

1.3.1 Topology information dissemination test

To test the flooding of domain information in the distributed architecture, information of a new (dummy) domain has been injected at one domain (viola-vidb-2). Table 1.2 shows the time offsets at which this new domain information has been added to the other domains' databases. The random time offsets are due to the fact that the advertisement of current topology information to peer domains is triggered 5 minutes after the last advertisement has been completed.

Time offset [s]	Domain
0	viola-vidb-2
109	i2cat-vidb-2
140	viola-vidb-1
398	i2cat-vidb-1
438	crc-vidb-2
677	crc-vidb-2

Table 1.2: Detailed results of the fixed reservation tests

Each domain has added this new domain information to its database and forwarded it to each of its peers exactly once. Duplicates have been detected and were discarded. Therefore, this test has shown that the topology dissemination mechanism is functional.

1.3.2 Fixed reservation tests

For the fixed reservation tests, an immediately starting reservation between two endpoints was made. After a delay of 20s, it was verified that the status of the reservation is ACTIVE. Then, an availability request for the same two endpoints and the same starting time was made and it was verified that the reply was ENDPOINTS_NOT_AVAILABLE.

Connection descriptor	Blocked endpoints	Alternative start time offset
i2cat	-	360s
viola	10.7.225.7 10.7.226.7	325s
crc	-	0s
i2cat-viola	10.7.226.7	360s
i2cat-crc	-	360s
viola-crc	10.7.226.7	329s

Table 1.3: Detailed results of the fixed reservation tests

These tests were basically successful, but have shown some minor deficiencies. For one, not all systems actually return a list of blocked endpoints, even if the result code is ENDPOINTS_NOT_AVAILABLE. Second, the CRC domain failed to report an alternative start time offset. However, it should be noted that this is not a required, but an optional feature.

1.3.3 Malleable reservation tests

For testing malleable reservations, two different test types were defined. Both create fixed reservations that block the shortest routes prior to requesting a malleable reservation. In the first case, the duration of the blocking reservations is short enough such that the malleable reservation can be made along the shortest route after the fixed reservations have expired. In the second case, the fixed reservations are active until the deadline of the requested malleable reservation, so that the malleable reservation has to take a longer route.

1.3.3.1 *Deferred malleable reservation tests*

For these tests, the blocking reservations were fixed reservations with 60 minutes duration that were started immediately. The earliest start time for the malleable reservations was set to 30 minutes after the start time of the blocking reservations, and the deadline was set to 90 minutes after the earliest start time.

Consequently, a time window of 60 minutes was left for the malleable reservations, and the earliest possible actual starting time of the malleable reservation is 30 minutes after the start time specified in the request. Since the requested data amount to be transferred was 1GB and the requested maximum bitrate was 1GBit/s, the actual duration necessary for the transfer was 800s, which fits into this time window well.

Connection descriptor	Involved domains	Start delay
i2cat		
viola	viola-chablis-a3 viola-vidb-1	30 minutes
crc	crc-vidb-1 crc-vidb-2	60 minutes
i2cat-viola	i2cat-vidb-2 viola-chablis-a3 viola-vidb-1	60 minutes
i2cat-crc	crc-vidb-1 i2cat-vidb-1 i2cat-vidb-2	60 minutes
viola-crc (1)	crc-vidb-1 crc-vidb-2 viola-vidb-1	0
viola-crc (2)	crc-vidb-1 crc-vidb-2 viola-vidb-1	60 minutes

Table 1.4: Detailed results of the deferred malleable reservation tests

Table 1.4 shows the domains involved in the malleable reservations for each of the test connections and the start delay with respect to the earliest start time specified in the malleable reservation request. It can be seen that this delay is 60 minutes in most cases, due to the fact that the algorithm for creating malleable reservations in the IDB code assumes an alternative start time offset of 60 minutes if a service is reported to be unavailable, but no appropriate alternative start time offset is specified in the reply. Obviously, only the systems in the domains at the University of Bonn respond with appropriate alternative start time offsets.

The reason for the start delay of 0 for the “viola-crc (1)” connection is an important conclusion from the tests that have been performed. On first sight, it seems that this result is not possible, because the only routes between the two endpoints of the test connection that traverse the domains listed in the table are blocked by the blocking connections for a start delay of 0. A closer look at the connections reveals the following: The IDB receiving the original request computes the following interdomain path:

> 10.7.226.7 – 10.7.227.5, 10.8.4.5 – 10.8.4.3, 10.8.3.3 – 10.8.3.1

Then, viola-vidb-1 checks the availability of 10.7.226.7 – 10.7.227.5. Since the two interdomain links between its ARGON subdomains are blocked, it finds an alternative path:

> 10.7.226.7 – 10.7.226.38, 10.7.225.38 – 10.7.225.37, 10.3.3.4 – 10.3.3.6, 10.3.4.6 – 10.3.4.4, 10.8.3.6 – 10.8.3.4, 10.8.4.4 – 10.8.4.38, 10.7.227.38 – 10.7.227.5

Therefore, the complete path that is established consists of 9 connections on NRPS level.

This is clearly an undesirable behaviour that is caused by the fact that the search for a feasible path should involve peer domains only at the highest hierarchy level and should then be restricted to subdomains once this level is left. WP1 is currently planning a solution for this problem.

Aside from this, the tests show that the malleable reservation handler is able to find an available time slot for a reservation, if resources are partly blocked by other reservations.

1.3.3.2 *Rerouted malleable reservation tests*

The test setup of the following tests is the same as for the tests described in the previous section, except for the duration of the blocking reservations which was extended to 120 minutes. Therefore, the malleable reservation could not be made on the shortest path.

Connection descriptor	Involved domains	Start delay
i2cat		
viola		
crc		
i2cat-viola	crc-vidb-1 crc-vidb-2 i2cat-vidb-1 i2cat-vidb-2 viola-vidb-1	0
i2cat-crc	crc-vidb-1 crc-vidb-2 i2cat-vidb-2 viola-chablis-a3 viola-vidb-1	0
viola-crc (1)	crc-vidb-1 crc-vidb-2 viola-vidb-1	0
viola-crc (2)	crc-vidb-1 i2cat-vidb-1 i2cat-vidb-2 viola-chablis-a3 viola-vidb-1	0

Table 1.5: Detailed results of the rerouted malleable reservation tests

Table 1.5 shows the results of these tests. For the same reason described in the previous section, the connection for “viola-crc (1)” seems to take a shortest path, which is not true because the requested connection between the endpoints in viola-vidb-1 is a multidomain connection traversing all of its peer domains.

These tests show that the malleable reservation is handler is able to use alternative routes when resources are blocked by other reservations.

2 Control plane experiments – WP2

This section provides the information and the conclusion about functional testing on the different modules implemented for the GMPLS and G^{2}MPLS Control Plane. It reports on the specific functional tests conducted to identify and confirms the proper operation of the various G²MPLS modules (developed from scratch or modified with respect to the Quagga software baseline) on real equipments deployed in some Phosphorus local test-beds (fiber and wavelength switches). The actual G^{2}MPLS Control Plane functional tests are been logically reported under three main sections: the LSP signalling tests, the call signalling tests and the routing tests.

2.1 Testing environment

This section describes the testing environment used to test the functionalities of the G^{2}MPLS Control Plane software, focusing on the physical topology, the Control Plane and the G^{2}MPLS stack configuration.

2.1.1 Transport Plane

This section reports the details of the PSNC and UESSEX test-beds used for the G^{2}MPLS Control Plane test. One fibre switched test-bed from UESSEX, one fibre switched test-bed from PSNC and one wavelength switched test-bed from PSNC were involved in the test. Details of the test-beds and the involved equipments are described in the following sections.

Table 2.1 shows equipment inventory in PSNC and UESSEX local test-beds.

EQUIPMENT					
PSNC			UESSEX		
Type/Make/ Model	**No.**	**Attrib.**	**Type Make/Model**	**No.**	**Attrib.**
ADVA FSP 3000RE-II (Lambda Switch)	3	15 Pass through ports 6 Local ports	CalientDiamondWave (Fibre Switch)	1 (4 after partitioning)	96 ports total
CalientDiamond Wave (Fibre Switch)	1 (4 after partitioning)	144 ports total	VLAN capable GE switch (FastIronFoundry)	1	24 ports optical
VLAN capable GE switch (XMR Foundry)	1	60 ports	VPN Gateway	1	----
Equipment controller	7	Virtualized PC	Equipment Controller	4	PCs
Client nodes	2	HP IA64 2xIntel Itanium2 servers	Client nodes	2	Intel dual core processor servers
Traffic Analyser/Generator (Agilent)	1	2X10GE 2x1GE	Traffic Analyser/Generator (Anritsu)	1	10GE network analyser

Table 2.1: Summary of available equipments in PSNC and UESSEX domains for G^2MPLS functional tests

2.1.2 Physical topology

The aforementioned equipments and facilities were used to build two test-beds in PSNC and UESSEX for the Control Plane functionality test. Topologies of these test-beds are described below:

2.1.2.1 *UESSEX fibre switched test-bed topology*

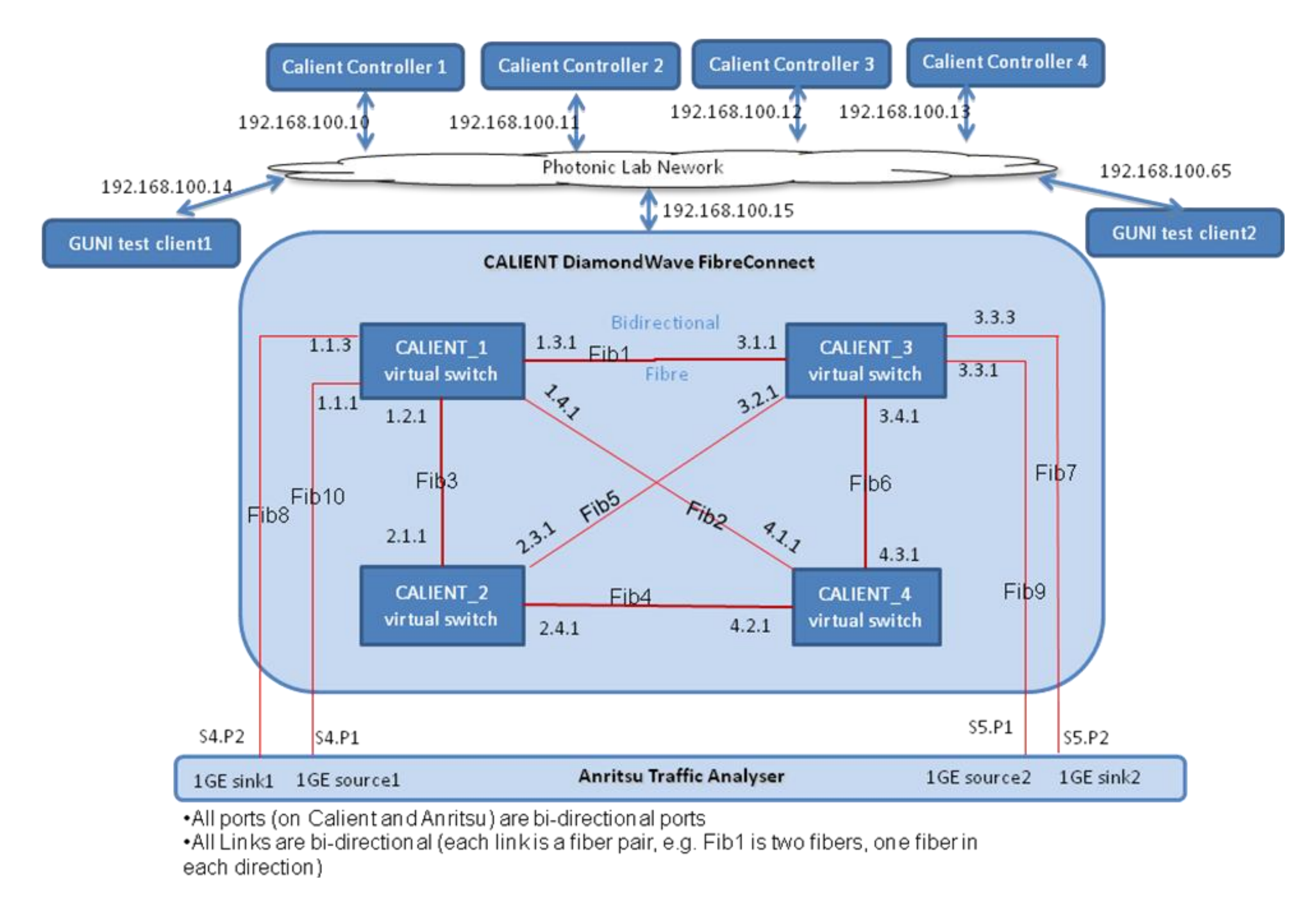

Figure 2.1: UESSEX fibre switching test-bed topology

The test-bed comprises one Calient Diamondwave Fiber Connect as the optical fibre-switching node. To emulate functionality of a realistic network with multiple optical switching nodes, the Calient switch has been partitioned into four independent sub- switches (i.e. Calient_1, Calient_2, Calient_3 and, Calient_4). This has been done through a proprietary software interface jointly developed by UESSEX and PSNC. Through the software interface, the Control Plane interfaces to four switches that can be communicated and operated independently. The four switches are interconnected with bi-directional optical fibres in a fully meshed topology. As shown in **Figure 2.1**, there is one direct bi-directional path between each two nodes. In the test-bed, each switch is controlled by a G^{2}MPLS node controller, which is an Intel quad core server and runs an instance of the G^{2}MPLS Control Plane. These four nodes are connected through the 1GE local area network, which constitute the Signalling Control Network (SCN) for the G^{2}MPLS Control Plane. In this test-bed, the user's Transport Plane is emulated by an Anritsu traffic analyser/generator. Each user uses one sink-source port of the traffic generator to transmit-receive data with optical 10GE format. For each user, there is an Inter dual core server that runs the G.UNI signalling and is connected to the SCN. **Figure 2.1** shows topology of the test-bed and its physical layer connectivity together with associated IP addresses, port numbers and fibre numbers.

2.1.2.2 *PSNC fibre switched test-bed topology*

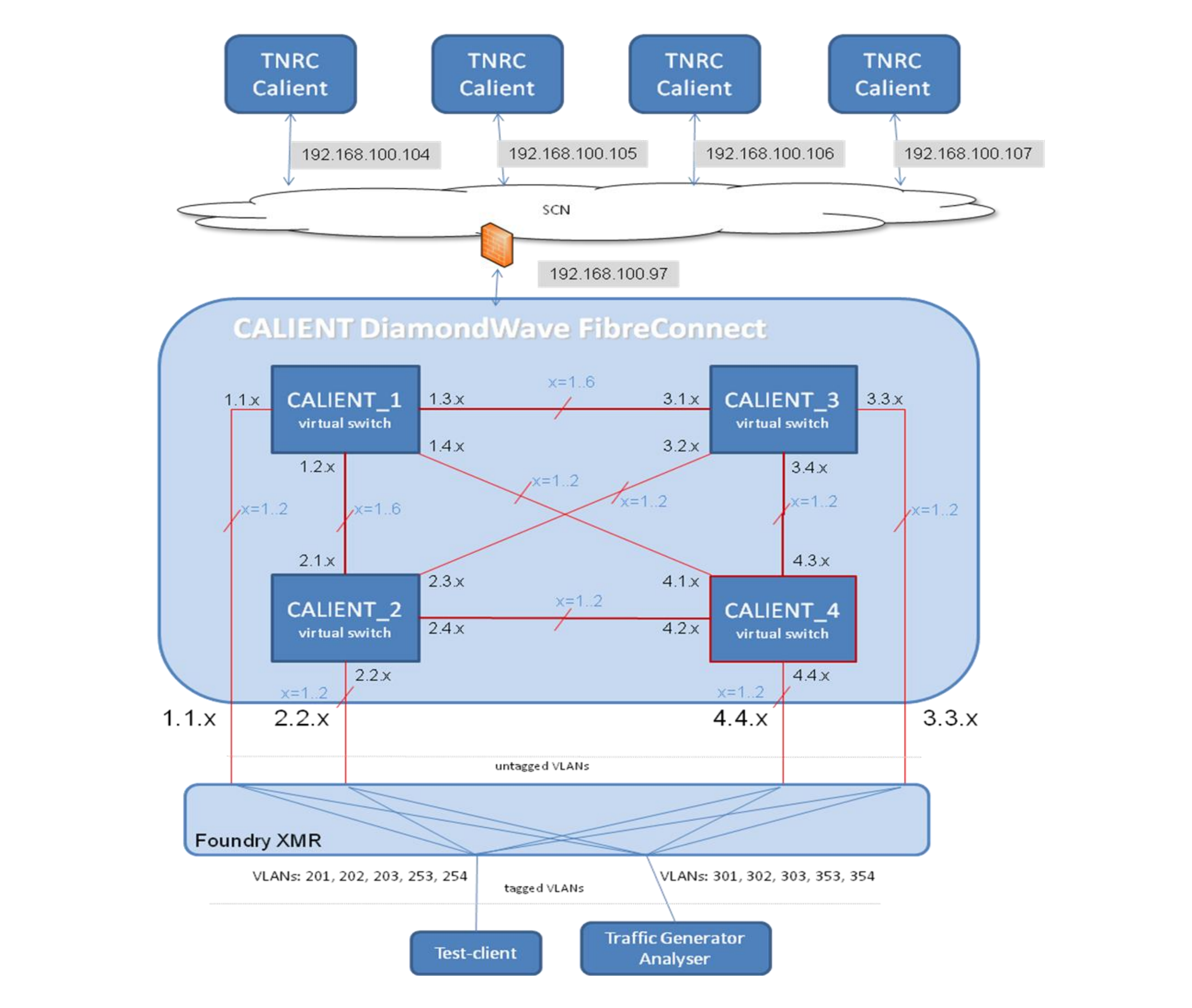

Figure 2.2: PSNC fibre switching test-bed topology

The test-bed comprises one Calient Diamondwave Fiber Connect as the optical fibre-switching node. To emulate functionality of a realistic network with multiple optical switching nodes, the Calient switch was partitioned into four independent sub-switches, as in the UESSEX case. The four switches are connected with bi-directional optical fibres in a fully meshed topology. As shown in Figure 3.5, there is 2-6 direct bi-directional path between each two virtual nodes. More data links between virtual nodes give great possibility to setup a set of LSPs going through the same transport plane node or setup LSPs using two or more data links as component links. In the test-bed, each switch is controlled by a G^2MPLS node controller (i.e. Calient controller 1..4), which is an Intel Itanium2 core server and runs an instance of the G^2MPLS stack These four nodes are connected through the local area network implememting the SCN. In this test-bed client nodes are emulated by a test client and a traffic analyser/generator. The client nodes are connected to the optical switches through a foundry XMR switch. Each client uses a range of VLAN tags as shown in Figure 3.5 to transmit-receive data with optical 1GEtherent format. Each VLAN tag is associated with fibre connectivity to one switching node. Each client also runs the client signalling or G.UNI and is connected to Control Plane network.

2.1.2.3 *PSNC wavelength switched test-bed topology*

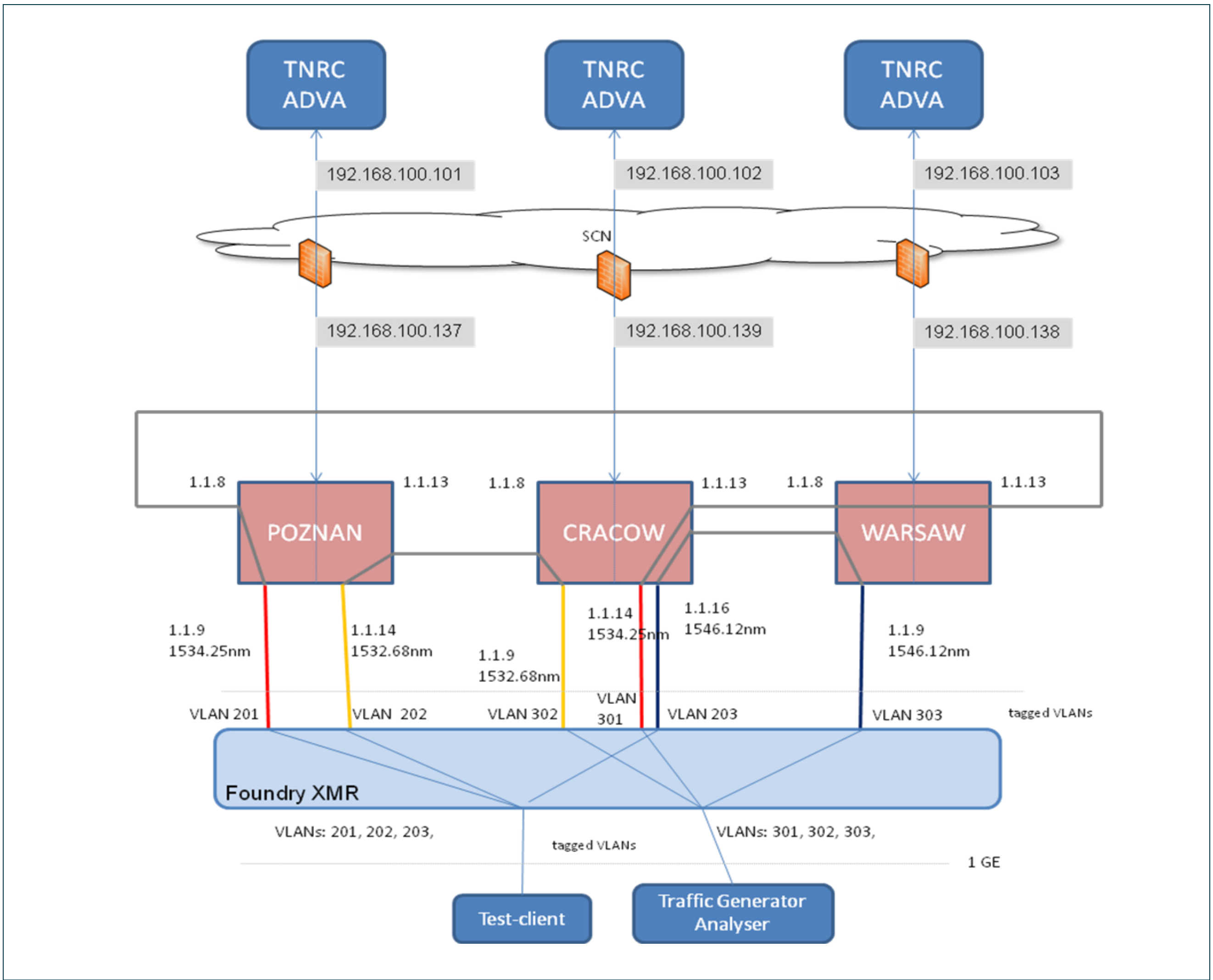

Figure 2.3: PSNC wavelength switching test-bed topology

The test-bed comprises three ADVA FSP 3000RE-II wavelength switches as the optical wavelength-switching nodes (POZNAN, CRACOW, WARSAW). The ROADMs are connected with bi-directional optical fibres in a ring topology. In the test-bed, each optical switch is controlled by a G^2MPLS node controller, which is an Intel quad core server. These three nodes are connected through the local area network implementing the SCN. In this test-bed client nodes are emulated by a test client and a traffic analyser/generator. The client nodes are connected to the optical switches through a foundry XMR switch. Each client uses a range of VLAN tags as shown in **Figure 2.3** to transmit-receive data with optical 1GE format. Each VLAN tag is associated to a specific fibre connectivity to one switching node. Each client also runs the G.UNI signalling and is connected to the SCN.

2.2 G^2MPLS Control Plane functional tests

2.2.1 Overview of the tests

The functionality of G^2MPLS Control Plane was tested using 35 test-cards divided into three main areas. Each test-card verifies a set of G^2MPLS features expressed as set of objectives to be achieved in the fixed environment conditions. All the test-cards have been successfully executed and thus the current state of the G^2MPLS Control Plane prototype development fulfil the main expected functionalities as per Milestone M2.4.

LSP signalling tests			
	LSP signalling tests in LSC domain		
No	**Test Card**	**Test name**	**Status**
1	G^2MPLS-TC-1.1	LSC node initialization	Passed
2	G^2MPLS-TC-1.2	Transport Plane notifications from LSC node	Passed
3	G^2MPLS-TC-1.3	Setup of one bidirectional LSC LSP	Passed
4	G^2MPLS-TC-1.4	Tear down of one bidirectional LSC LSP from HEAD node	Passed
5	G^2MPLS-TC-1.5	Tear down of one bidirectional LSC LSP from TAIL node	Passed
6	G^2MPLS-TC-1.6	Unsuccessful bidirectional LSC LSP setup (failure in HEAD node)	Passed
7	G^2MPLS-TC-1.7	Unsuccessful bidirectional LSC LSP setup (failure in intermediate node)	Passed
8	G^2MPLS-TC-1.8	Unsuccessful bidirectional LSC LSP setup (failure in TAIL node)	Passed
9	G^2MPLS-TC-1.9	Setup of one bidirectional LSC LSP with advance reservation	Passed
10	G^2MPLS-TC-1.10	Tear down of one bidirectional LSC LSP with advance reservation from HEAD node	Passed
	LSP signalling tests in FSC domain		
No	**Test Card**	**Test name**	**Status**
11	G^2MPLS-TC-2.1	FSC node initialization	Passed
12	G^2MPLS-TC-2.2	Transport Plane notifications from FSC node	Passed
13	G^2MPLS-TC-2.3	Setup of one bidirectional FSC LSP	Passed
14	G^2MPLS-TC-2.4	Tear down of one bidirectional FSC LSP from HEAD node	Passed
15	G^2MPLS-TC-2.5	Tear down of one bidirectional FSC LSP from TAIL node	Passed
16	G^2MPLS-TC-2.6	Unsuccessful bidirectional FSC LSP setup (failure in HEAD node)	Passed
17	G^2MPLS-TC-2.7	Unsuccessful bidirectional FSC LSP setup (failure in intermediate node)	Passed
18	G^2MPLS-TC-2.8	Unsuccessful bidirectional FSC LSP setup (failure in TAIL node)	Passed
19	G^2MPLS-TC-2.9	Setup of one bidirectional FSC LSP with advance reservation	Passed
20	G^2MPLS-TC-2.10	Tear down of one bidirectional FSC LSP with advance reservation from	Passed

No	Test Card	Test name	Status
		HEAD node	
G^2MPLS call signalling tests			
	Single-domain call signalling tests		
No	**Test Card**	**Test name**	**Status**
21	G^2MPLS-TC-3.1	Setup of one bidirectional single-domain FSC LSP by G2.NCC module	Passed
22	G^2MPLS-TC-3.2	Teardown of the one bidirectional single-domain FSC LSP by G2.NCC module	Passed
23	G^2MPLS-TC-3.3	Setup of one bidirectional single-domain FSC LSP by G2.CCC module	Passed
24	G^2MPLS-TC-3.4	Teardown of the one bidirectional single-domain FSC LSP by G2.CCC module	Passed
25	G^2MPLS-TC-3.5	Setup of one bidirectional single-domain FSC LSP by G.UNI-GW module	Passed
26	G^2MPLS-TC-3.6	Teardown of the one bidirectional single-domain FSC LSP by G.UNI-GW module	Passed
27	G^2MPLS-TC-3.7	Setup of one bidirectional single-domain FSC LSP by Middleware WS-Agreement client	Passed
28	G^2MPLS-TC-3.8	Teardown of the one bidirectional single-domain FSC LSP by Middleware WS-Agreement client	Passed
	Inter-domain call signalling tests		
No	**Test Card**	**Test name**	**Status**
29	G^2MPLS-TC-4.1	Setup of one bidirectional inter-domain FSC LSP by G2.CCC	Passed
30	G^2MPLS-TC-4.2	Teardown of the one bidirectional single-domain FSC LSP by G2.CCC	Passed
G^2MPLS routing tests			
	Single-domain routing test-cases		
No	**Test Card**	**Test name**	**Status**
31	G^2MPLS-TC-5.1	I-NNI G2.OSPF-TE instance initialization	Passed
32	G^2MPLS-TC-5.2	Distribution of TE information through the G.I-NNI interfaces	Passed
33	G^2MPLS-TC-5.3	Distribution of Grid information through the G.UNI and G.I-NNI interfaces	Passed
	Inter-domain routing test cases		
No	**Test Card**	**Test name**	**Status**
34	G^2MPLS-TC-6.1	Routing information exchange between adjacent RAs	Passed
35	G^2MPLS-TC-6.2	Grid information exchange between adjacent RAs	Passed

Table 2.2: Overview of the executed test-cards

2.2.2 LSP signalling tests

The G^2MPLS LSP signalling tests have been executed in two separate sessions:

- LSC LSP signalling tests
- FSC LSP signalling tests

The LSC LSP signalling tests have been used to verify the proper work and interaction of modules involved in the LSP signalling in LSC domain and LSC equipment configuration (G^2.RSVP-TE, LRM, TNRC with Adva TNRC SP plugin, SCNGW).
Similarly, The FSC LSP signalling tests have been used to verify the proper work and interaction of modules involved in the LSP signalling in FSC domain and FSC equipment configuration (G^2.RSVP-TE, LRM, TNRC with Calient TNRC SP plugin, SCNGW).

Figure 2.4 shows the single-domain LSP signalling in LSC domain. In the test-bed there are 3 G^{2}MPLS controllers with just I-NNI interfaces between them.

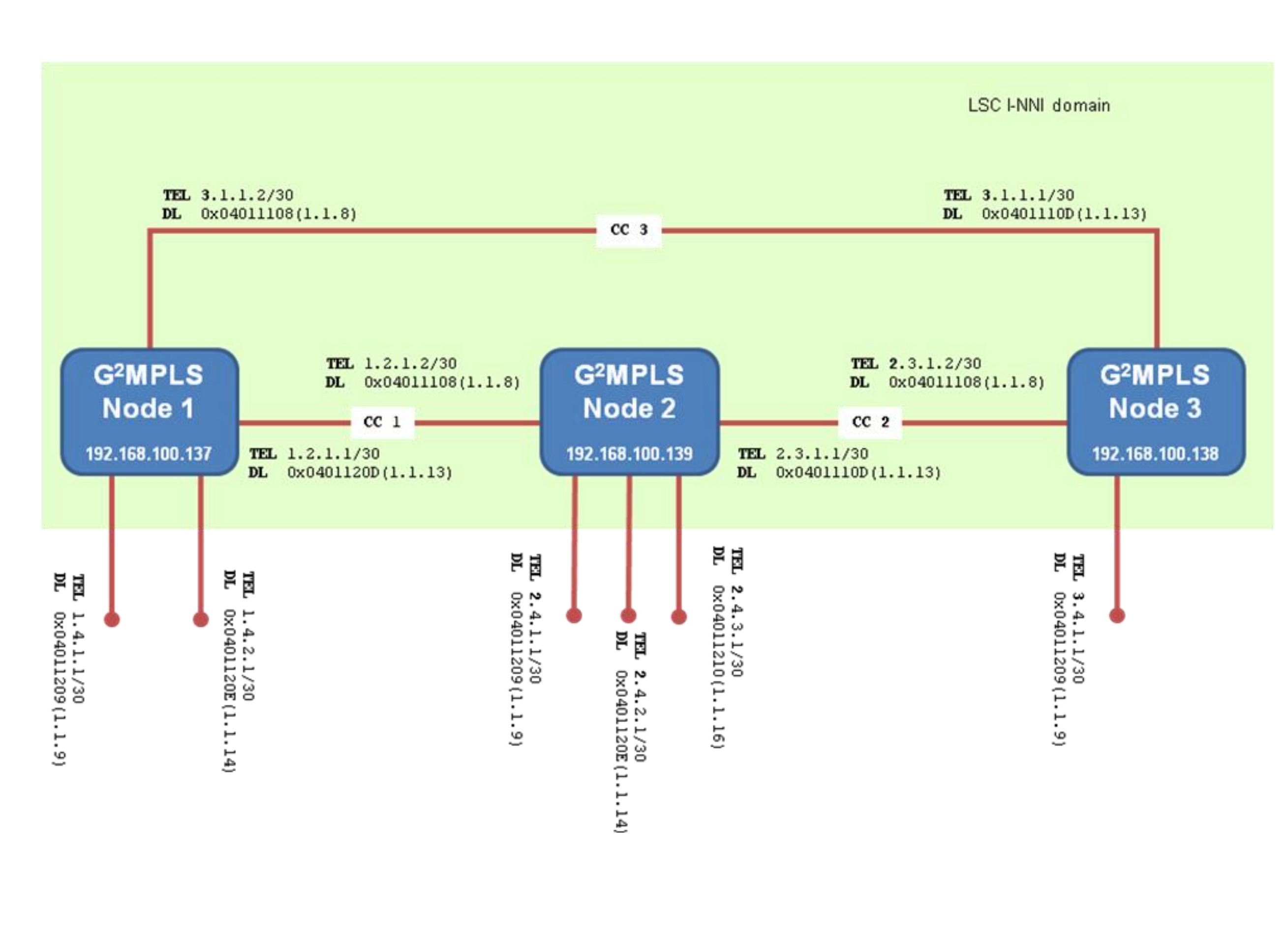

Figure 2.4: Logical topology of the single-domain LSC test-bed

Figure 2.5 shows the single-domain LSP signalling with FSC switching capability. In the test-bed there are 4 I-NNI G^{2}MPLS controllers.

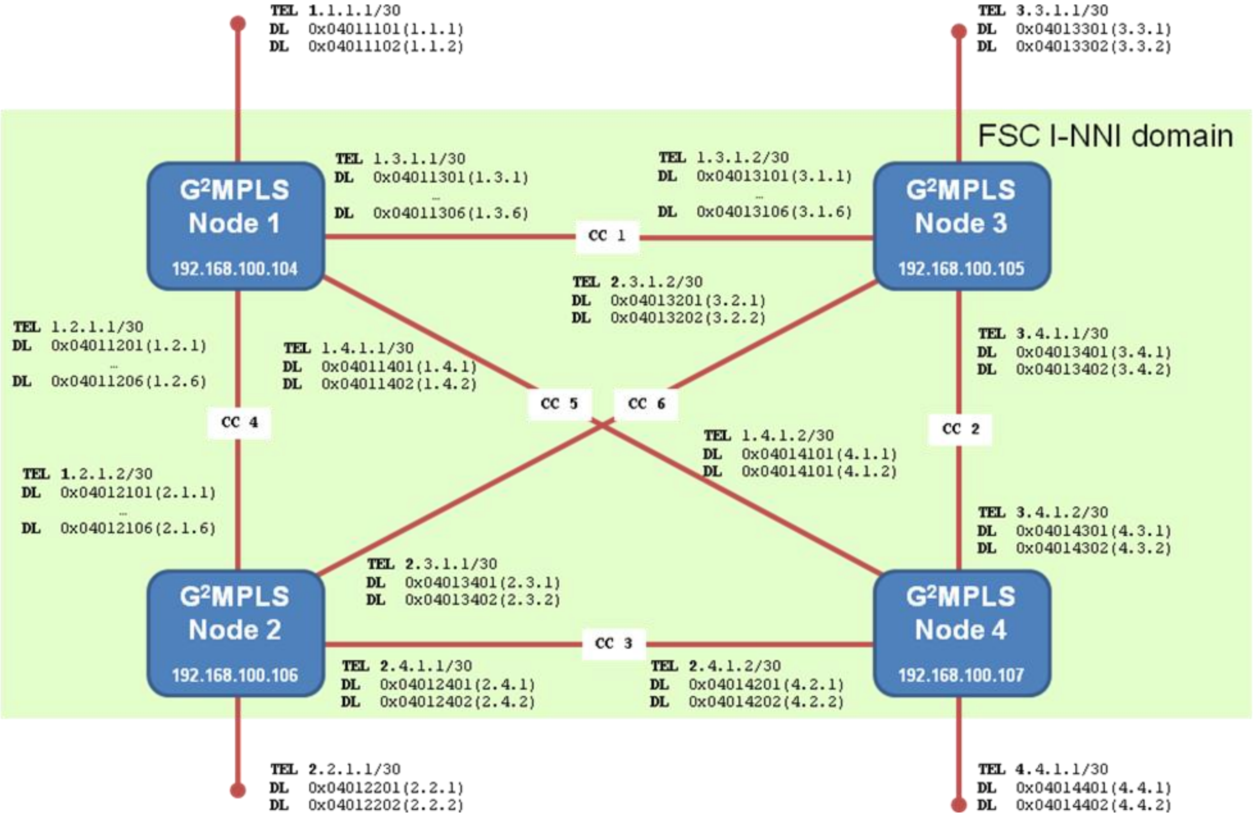

Figure 2.5: Logical topology of the single-domain FSC test-bed

2.2.3 G^{2}MPLS call signalling tests

The G^{2}MPLS call signalling tests have been executed in two separate sessions:

- single-domain call signalling tests
- Inter-domain call signalling tests

The single-domain tests have been used to verify the proper work and interaction of that modules involved in just one G^{2}MPLS domain signalling (mainly G2.NCC, RC and G^2.RSVP-TE).
The inter-domain tests, instead, have been used to verify the proper work and interaction of those modules involved in the multi-domain signalling (mainly G2.NCC and G.ENNI-RSVP).

Figure 2.6 shows the single-domain call signalling tesbed. In the test-bed there are 6 different G^{2}MPLS controllers: 4 of them (Node 1, 2, 3 and 4) have been used as INNI nodes, and the other 2 as G.UNI-GW clients.

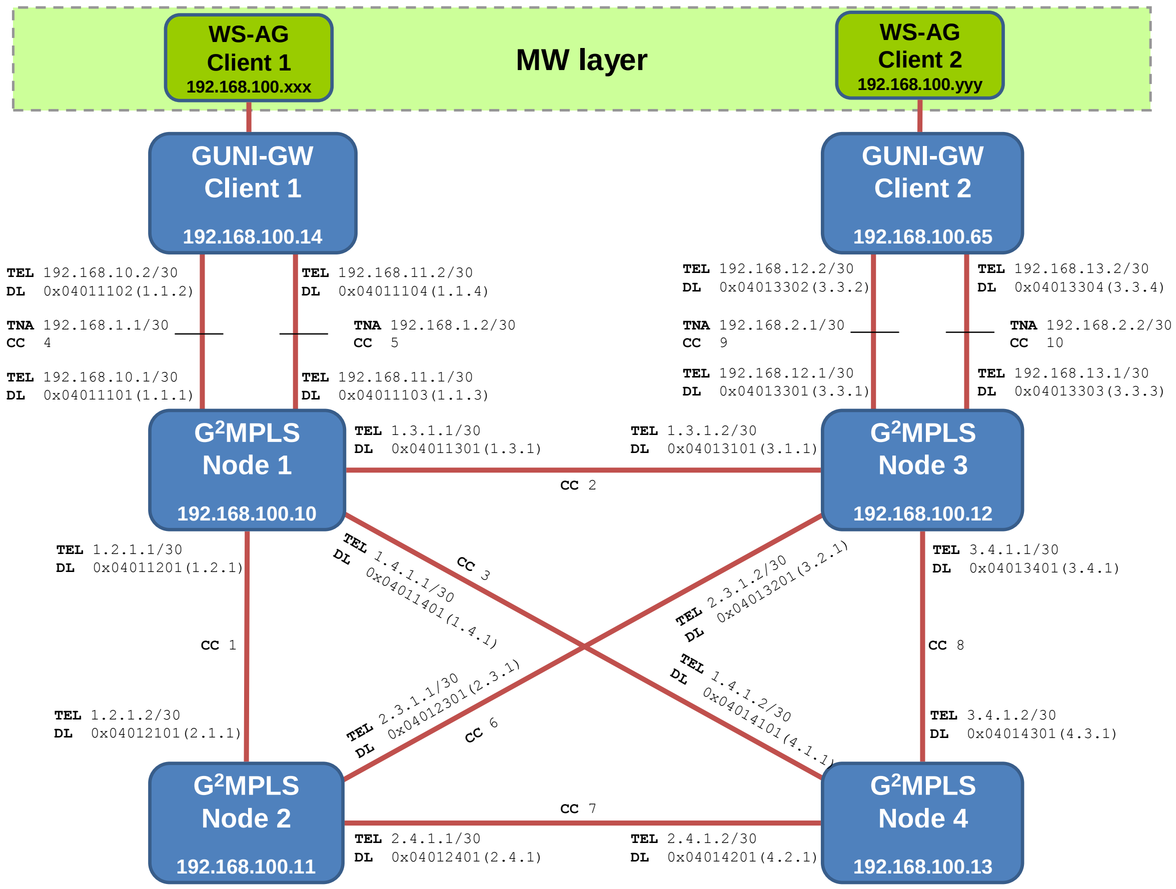

Figure 2.6: Logical topology of the single-domain FSC test-bed for G2MPLS Call signalling tests

In **Figure 2.7** the logical topology of the test-bed for the inter-domain call signalling tests is described. For the purpose of the tests, 6 G^{2}MPLS controllers have been deployed, in order to setup 2 different I-NNI domains (each one with 2 controllers) reachable by a G.UNI-GW client each.

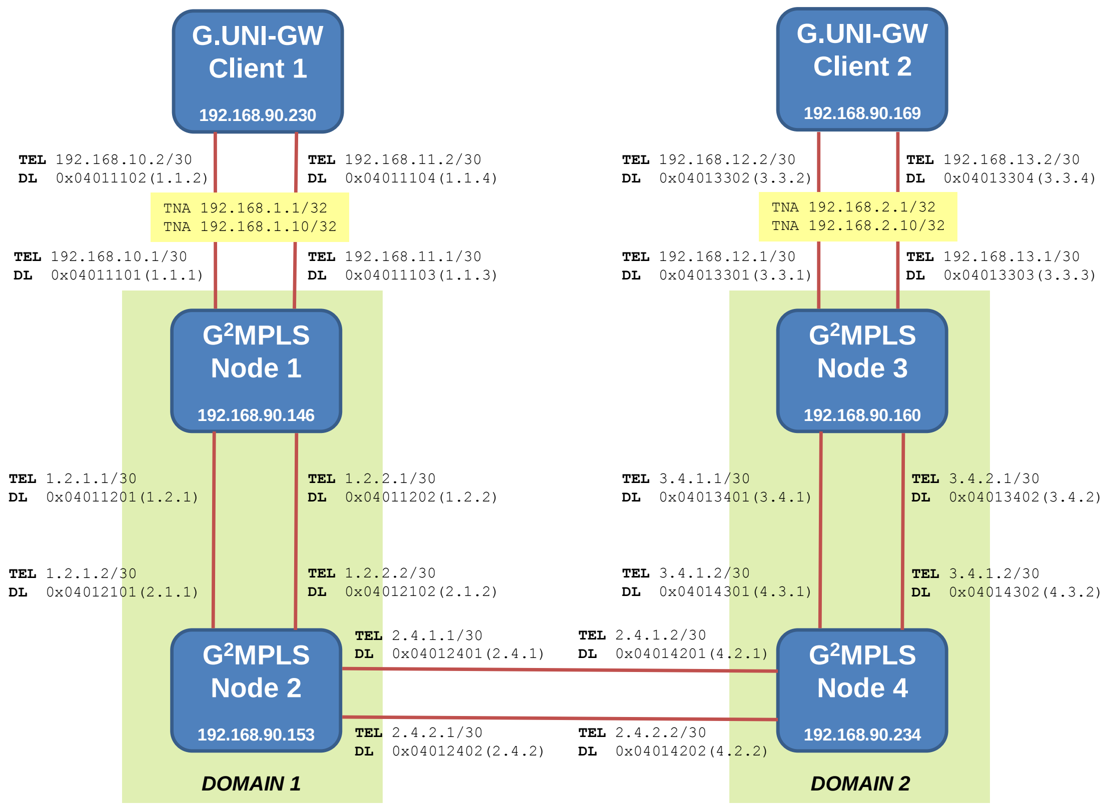

Figure 2.7: Logical topology of the inter-domain FSC test-bed for G^{2}MPLS Call signalling tests

2.2.4 G^{2}MPLS routing tests

The G^{2}MPLS routing advertisement tests have been executed in two separate sessions:

- G^{2}MPLS single-domain routing tests
- G^{2}MPLS Inter-domain routing tests

The single-domain tests have been used to verify the proper work and interaction of that modules involved in the single-domain routing (mainly G2.OSPF-INNI, G2.OSPF-UNI, LRM, SCNGW).
The inter-domain tests instead have been used to verify the proper operation and interaction of those modules involved in the multi-domain routing (mainly G2.OSPF-INNI, G2.OSPF-ENNI (referred also as ENNI-RC), G2.OSPF-UNI, LRM, SCNGW).

The G^2MPLS routing tests are divided also regarding network/grid functionalities:

- Routing network resources advertisement tests
- Routing Grid&Network resources advertisement tests

Routing network resources advertisement tests check the single-domain and multi-domain advertisement of router addresses, TE-links and TNA addresses. These test verify compatibility with standard GMPLS.
Routing Grid&Network resources advertisement tests check the single-domain and multi-domain advertisement of Grid resources in parallel to network resource advertisement.

Figure 2.8 shows the single-domain routing advertisement testbed. In the test-bed there are 6 different G^2MPLS controllers: 4 of them (Node 1, 2, 3 and 4) have been used as INNI nodes, and the other 2 as G.UNI-C clients.

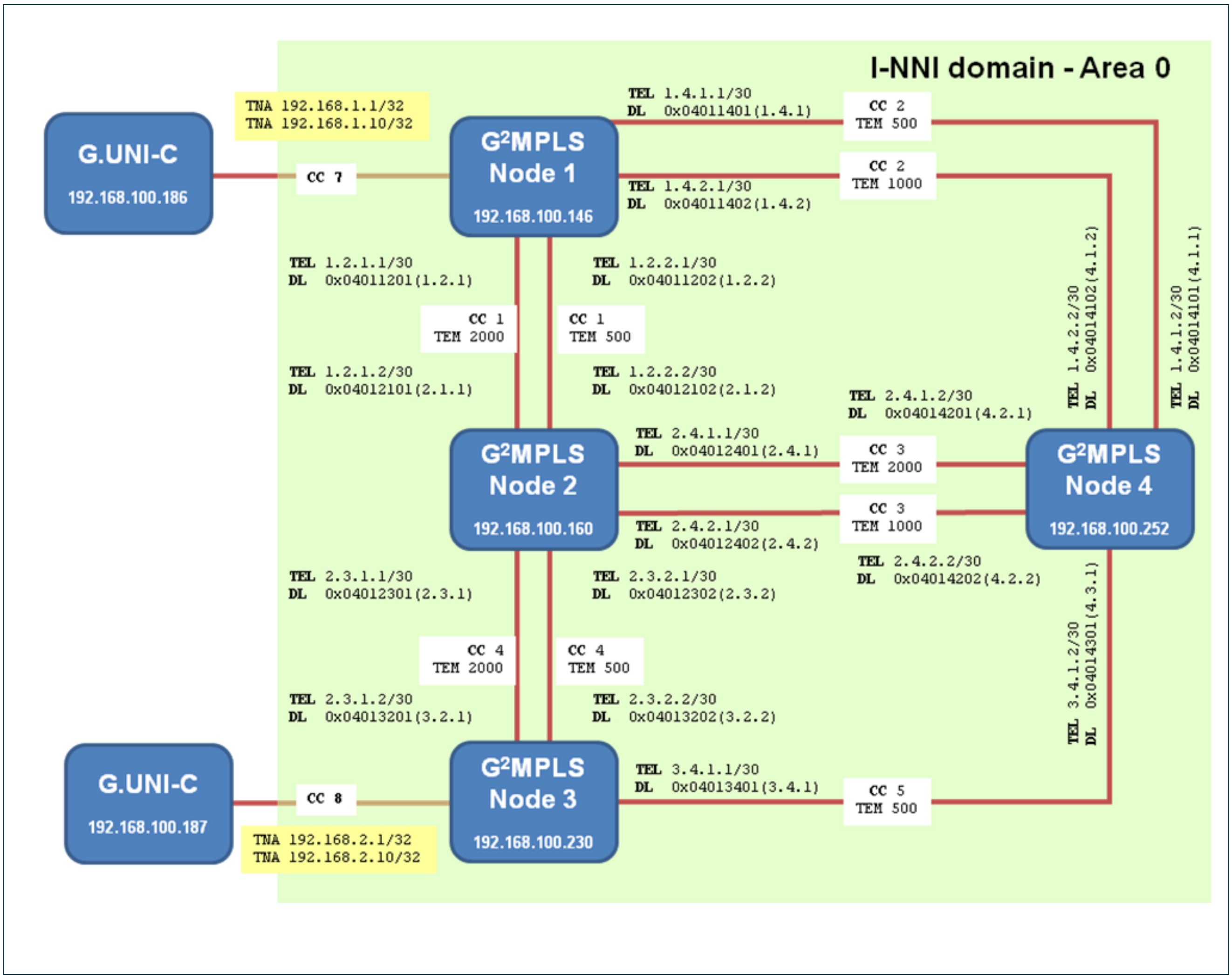

Figure 2.8: Single-domain logical topology for routing tests

In Figure 2.9 the logical topology of the test-bed for the inter-domain routing advertisement tests is described. For the purpose of the test there is need for at least 6 G^{2}MPLS controllers, to have 2 different I-NNI domains (each one with 2 controllers) reachable by a G.UNI-C each.

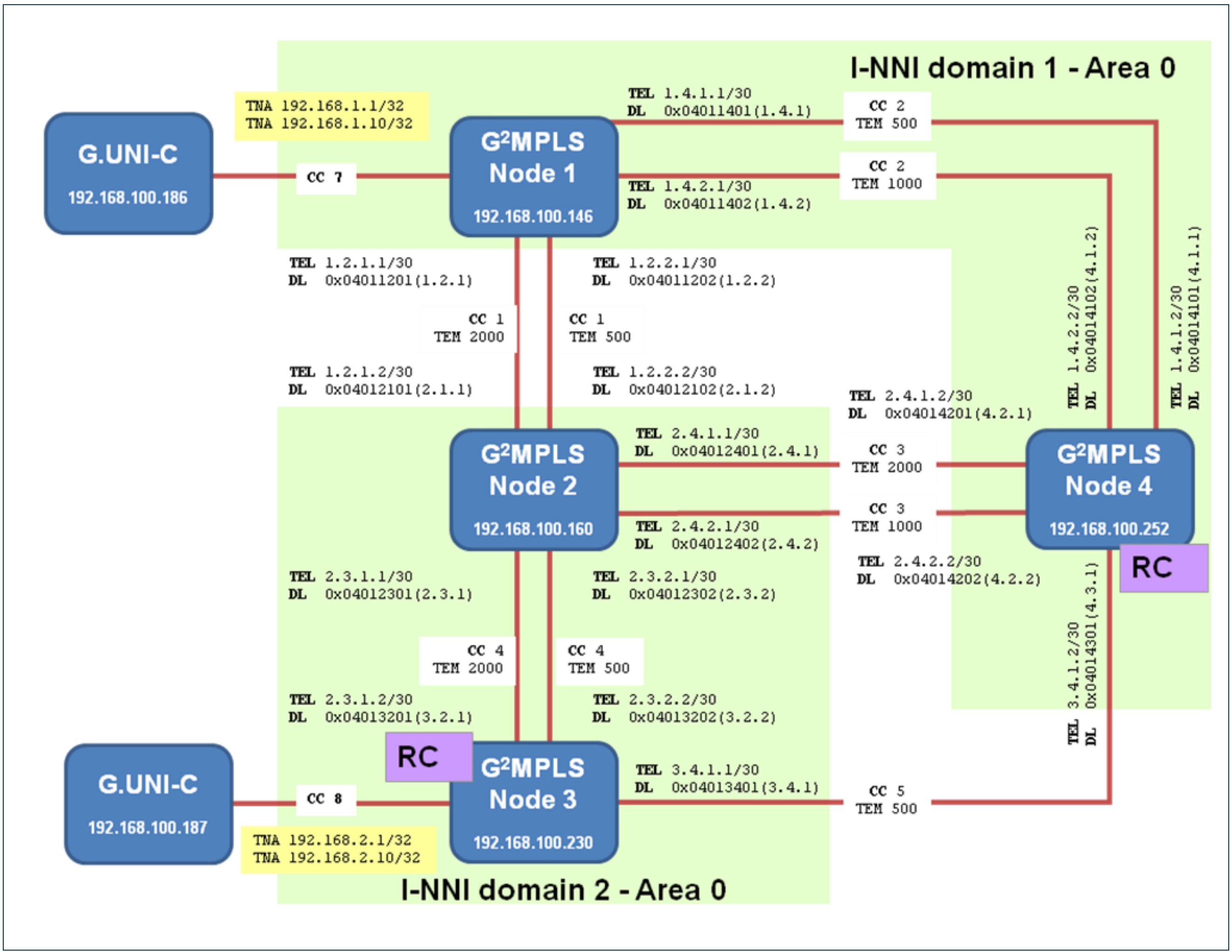

Figure 2.9: Inter-domain logical topology for routing tests

2.3 G²MPLS Test Conclusions

This section reports the environment for the functional tests on the G²MPLS Control Plane built in the framework of Phosphorus WP2. Most of these tests comprised the single module and functionality verification, as well as the integrated operation of protocols and network controllers in meshed topologies. These test were important for public release of the G²MPLS Control Plane. They represent a first step of G²MPLS integration in the Phosphorus test-beds, an activity that will be progressed and finalized by WP6 team. In the PSNC and UESSEX local test-beds two switching technologies have been deployed, i.e. the fiber and the lambda switching, and proper mediation modules have been developed between the G²MPLS protocols and the Calient Diamond Wave Fibre Connect and the ADVA FSP-3000RE-II switches.

All the planned tests have been concluded successfully, both in the single-domain and in the inter-domain cases. They can be used as a test-suite for testing G^2MPLS prototypes in different deployment scenarios., e.g. other G^2MPLS test-beds installed in Phosphorus framework or externally. The same test-suite can help the validation phase of new developments on the stack, e.g. in case of addition of other mediation functions towards other equipments or switching technologies.

3 Middleware and applications experiments – WP3

The developments of WP3 are directly targeted to perform test-bed experiments. The middleware will make the Phosphorus network services accessible to applications and the applications are modified and enhanced to make use of those services. Following the two phases of the project developments, there are two phases for middleware and application experiments. In the current phase of the project, the WP3 experiments a have focussing on two fields

- Test and evaluate the applications in the PHOSPHORUS middleware framework
  - TOPS
  - WISDOM
  - DDSS
  - KoDaVis
- Test and evaluate the middleware extensions
  - Test and evaluate the interaction of middleware and the underlying services as provided by WP2.

Besides these experiments WP3 also tested smaller changes in the middleware (MetaScheduling Service and UNICORE) that had to be implemented in the course of the experiments.

A detailed discussion of the application experiments can be found in the deliverable D3.6 “Report on the Results of the Application Experiments During the Final Testbed Experiments”. The middleware experiments are described in more detail in D3.7 “Report on the results of the middleware experiments during the final testbed experiments”.

3.1 Application experiments

- **TOPS**

TOPS (Technology for Optical Pixel Streaming) enables remote viewing of large scientific datasets (2D or 3D) on high resolution display devices (Tiled Panel Displays). TOPS streams these pixels, uncompressed, from the data center over the network to remote displays.

TOPS data center and the rendering machines are located in Amsterdam, the resulting picture is streamed over the network to Sankt Augustin and be viewed on the i-CONE™ display of FhG IAIS. At the display site the scientist interacts (navigates through the data set in real-time) with the application that runs at the rendering site. The i-CONE™ is a cylindrical 270-degree projection display system with high-resolution and evenly curved projection surfaces. The i-CONE™ has a visitor capacity of approximately 20 people. The display consists of four projectors with a resolution of 1600x1460 pixel at vertical refresh rate of at 105 Hz each. The projectors support active stereo mode which means that for each eye half of the refresh rate is available. The input for the projectors is provided by a cluster of four workstations equipped with NVIDIA Quadro FX graphics cards and a gigabit network interface each.

In the final phase of the test bed experiments, SARA and FhG tested TOPS over the connection SARA-Switch-Geant2+-Viola-FhG. On this connection a dedicated 1 Gbps lightpath is available for TOPS. As was indicated in deliverable D3.4, modifications were required to adapt TOPS to the I-Cone display at FhG. Since then, configuration of TOPS has been parameterized to allow for arbitrary display formats.

Monitoring of the performance of TOPS was done on the SURFnet provided test bed switch, located at SARA and connected to Geant2+ on one side and to number of PHOSPHORUS partners via NetherLight. The SURFnet NOC, executed by SARA, controls and monitors the Phosphors switch.

It could be shown that TOPS is able to use the complete bandwidth that was made available for the experiments during the whole timeframe. This indicates that the use of pan-European, multi-domain lightpaths will allow for continuous availability of required bandwidth and quality of service for high performance applications.

Although latency was not measured quantitatively, the tests demonstrated that with the available bandwidth and the configuration of the lightpaths, the latency is minimal and very acceptable for the end-user application.

- **WISDOM**

After the inspection of the first EGEE Data Challenge and their WISDOM workflow necessary changes for the PHOSPHORUS environment were identified. As a result of this analysis different WISDOM workflow phases were identified, which must be enabled based on MSS and UNICORE 6: Stage-in, execution and stage-out phases for each WISDOM job.

For the first test phase of WISDOM a test data suite from BioSolveIT was selected. Based on local installations tests were done on all four PHOSPHORUS sites to test the different docking software modules.

For WISDOM application tests the BioSolveit flexx-200 Testdata suite was selected and adapted to the different data formats of both the applications AutoDock and FlexX. The BioSolveit flexx-200 Testdata suite is a subset from the PDB, where each ligand was separated from the protein-ligand-complex and hand-fixed concerning protonation, aromaticity, delocalisation, and formal charges. Additionally start scripts and data environments were modified to realise a good basis for UNICORE 6 executions. Finally code installations and test data sets were made available on all participating PHOSPHORUS sites (except FlexX at PSNC, because of operating system incompatibility).

Besides the implementation of efficient workflows the major focus during this phase was on optimisation of the data transfer (state-in and state-out parts of the workflow).

On the technical side, UNICORE 6 complies with the OASIS WSRF 1.2 and OGF JSDL 1.0 standards, provides pluggable file transfer mechanisms with the OGSA ByteIO standard as default. The usual way of transferring a file is, that first the client sends a SOAP message to the server, which initiates a file transfer object. The client gets as an answer a link to this object, thus the client is enabled for direct accesses. Via methods of the object further information and data can be exchanged. After termination of the file transfer, the link will be deleted. This procedure is repeated for each file transfer operation.

From the user perspective special job storage locations are used. If a user creates a job, he gets a link to the generated objects and is enabled to transfer files from the client or from storage in the Grid. For each job a so called User Space (USpace) is generated by the UNICORE system, which contains programs, scripts and other data. After termination of the job output data will be exported back to the client or specified remote storage locations.

UNICORE-ByteIO is the standard method for file transfer in UNICORE 6. It uses the feature in UNICORE to transfer information via web services. Thus ByteIO needs no special connections between the sites. All data will be send in small portions with SOAP messages via HTTP to the gateways, which forward again via HTTP to the target systems. In this way ByteIO uses automatically the encoding and authentication methods of UNICORE. UNICORE file transfer is based on one basis class, which offers functionality of a web service and exchange of data and error handling. For each web service a specific class for the client exists, which executes the exchange of SOAP messages. This is the basis for different file transfer classes, whereas in practice only the ByteIO method is used. Thus to resume: This is a simple, flexible and secure method, which encapsulate transferred data in SOAP messages. Data is following the same encoded way between client, gateway and server like the control information. Unfortunately this method lacks performance and reaches max. 400 KB/s. Additionally, UNICORE 6 is offering Basic File Transfer (BFT), which uses HTTP for file transfer and achieves network transfer performance of several MB/s.

Depending on the availability of a new, improved high performance data transfer method (called UDT) more experiments will be done at a later stage of the project. However, the performance data reported in an internal

FZJ report by T. Oistrez based on a prototype implementation of the UDT method compared with Globus' GridFTP are quite promising:

- GridFTP is based on the FTP-based data transfer protokoll. Data is exchanged between Server and Client/Server via multiple TCP connection. Different to normal FTP GridFTP send data encoded and uses more than one data channel. The middleware Globus uses GridFTP for data transfer in Grids.

- UDT uses the UDP-Hole-Punching method. The User Datagram Protocol (UDP) is a non reliable and non connection oriented transport layer protocol. The application that uses UDP has to make sure that the data is completely transmitted. Although no connections exist firewalls use a simple mechanism to feign a connection. The client generates the UDP datagram and sends it to the firewall. The firewall examines the datagram and forwards it to the destination. The concept of UDP hole punching can be easily modified to be used in Grid environments.

First results show improved performance using UDT, which is approximately three times faster than Globus GridFTP.

- **DDSS**

Two kinds of applications have been used in DDSS experiments. Testing of each application has been split into two tests scenarios:

- big files,
- lots of small files.

In case of DDSS GridFTP tests, the values of two parameters have been modified in each of the phases, both having the influence on the data transfer speed:

- number of threads (parallel data streams),

- block size.

In case of DDSS B/A tests, two parameters have been modified in each of the phases:

- TCPW (TCP Window Size),

- TCPBUFSIZE (TCP Buffer Size).

Actually, four various test scenarios were prepared for each application,

The range of parameters values used for testing was determined empirically. Prior to main testing, a few test rounds with the applications and based on the tests results were ran, and the range of the parameters values used for further testing we chosen.

The main experiments were controlled by the test automation tools were developed for this purpose. The results of experiments were placed on the web page [www1][www2] hosted in PSNC. The experiments results are divided into two groups: GridFTP and TSM B/A tests.

In case of DDSS GridFTP, the experiments were run twice: the first time on the Internet and then on the PHOSPHORUS infrastructure. In that way, we were able to aggregate and compare both tests results. The transmissions were run between PSNC and three other locations: FZJ, FHG and UEssex.

The DDSS B/A experiments were run only over the PSNC-FZJ link, as only two backup/archive servers (one in PSNC, and the other in FZJ) were assigned to PHOSPHORUS tests. Moreover, the experiments were also run locally in PSNC, over a LAN, using the PHOSPHORUS test TSM server and the local client node. The results of local tests are used for comparison with those performed over the PHOSPHORUS link.

The DDSS GridFTP tests results show that the PHOSPHORUS optical network links reservation features make significant improvement of the performance of large data transmission possible. From the results of the big files transmission, it can be seen that PHOSPHORUS reservation features can provide a high transmission bandwidth to the data transfer application. From this part of the testing scenarios, it can be derived that using a high level of parallelism one may overcome the transmission limits caused by high latency in both PHOSPHORUS and Internet setup. However, it can also be observed that, the high bandwidth assured by the PHOSPHORUS reservation features guarantee significantly higher transfer rates than the Internet link can provide, nevertheless of the number of the transmission threads.

Similarly, the DDSS Backup/Archive tests results show, that the PHOSPHORUS links reservation help to gain good performance for massive data transmissions. The numbers acquired in big files backup scenarios are comparable to results of reference tests performed in the local setup, over the LAN network in PSNC. This confirms the efficiency of bandwidth reservation functionality provided by PHOSPHORUS network.

- **KoDaVis**

One of the findings during the first phase of testbed experiments with the KoDaVis application was the lack of information about poorly performing links or partners in a collaborative session. During a session, it was intractable to determine the cause of problems, if they occured. Individual participants in a collaborative visualisation session can limit the session's performance. Therefore, monitoring capabilities were added to various components of the KoDaVis application. These monitoring capabilities are comprised of bandwidth monitoring of the connections between the data server and each of the clients. This is done in the visualisation client and only helps the local user to see problems with his connection. Additionally, the collaboration and data servers keep track of the transfer times whenever bursts of data are transferred to the individual clients. This performance data is then queried by the UNICORE KoDaVis service, exposed in its properties and thus available in the KoDaVis UNICORE plugin. The latter has been extended to display the performance data of the individual server to client connections.

While the general setup of KoDaVis hasn't changed from the initial reports on the architecture, it is worth noting that the interface between the "ADS: Visit" service, which is embedded in UNICORE and the collaboration and data servers has been greatly enhanced to allow for flexible performance monitoring.

It was already mentioned in the report about the first testbed experiments, that the serial version of the data server poses a bottle neck as the number of participants in a collaborative session grows. At that time, the parallel version of the data server was still immature and could not be used to improve performance. This has changed since, and better results can now be achieved through exploiting the parallel data server.

3.2 Middleware experiments

The focus of the experiments during the last period was on testing and evaluating the interaction of middleware and the underlying services as provided by WP2.

Our first remote tests in both directions were aimed on basic connectivity between Essex and Fraunhofer. We created several predefined dummy requests and responses to test general web service connectivity between our test sites. We accomplished those tests without problems.

In second stage of test phase, we tested submitting and processing of meaningful data between Fraunhofer SCAI and Essex.

As for GRR, functionality for requesting information about sites worked fine in both directions. Requests using `getResourcesOperation()` returned correct response. This function returns a array of GLUE documents. GLUE documents are stored on server. Sending and storing GLUE documents did also work without problems. `publishOperation()` was correctly transmitting and storing GLUE documents in our test bed. This function transmits a GLUE document to GRRService to store it. As for BESFactory, there was more functionality to test. After connectivity tests, we
concentrated on basic functionality of functions like: `createActivity()` and `terminateActivities()`. Missing BES functionality was added through extensions. Extensions have been used for adding features like start-, end time or network bandwidth. Functions for creating and terminating jobs were correctly transmitting and parsing proper values.

After those tests , we had to replace our web stack for BES and GRR because of merging with a different piece of code. Basic connectivity is restored now. Our previous tests are also running smooth. We still missing BES interface tests besides createActivity() and terminateActivities() .

In addition, several experiments have been performed successfully in order to prepare the demonstration at the SC08 and the ICT event.

<u>Local test-beds involved:</u>

- PSNC (applications deployed: DDSS, WISDOM, KoDaVis)
- VIOLA (applications deployed: TOPS, DDSS, WISDOM, KoDaVis)
- UESSEX (applications deployed: DDSS, WISDOM, KoDaVis)

– SARA (applications deployed: TOPS)

4 Authentication, Authorisation and Accounting experiments – WP4

The following subchapters describe both the WP1 interdomain data and control plane configuration used for the implemented WP1 Harmony AAA/AuthZ scenarios. This includes the topology information, naming spaces and addressing schemes. Afterwards the AAA/AuthZ scenarios themselves are depicted and the experiences received from the integration of the GAAA-TK library are given.

4.1 Interdomain data plane configuration

4.1.1 Interdomain connections: VLAN naming spaces

The different domains within the WP1 data plane are connected either directly, like the different VIOLA domains, or via L2VPNs. Interdomain links based on L2VPNs use either Géant2 point-to-point or connections within GLIF infrastructure.

The domain numbering convention was defined and agreed as follows:

Domain/Site numeric identifier	Domain Name
1	PSNC
2	*not used (CESnet)*
3	I2CAT
4	SURFnet
5	SARA
6	UESSEX
7	VIOLA
8	CRC
9	Internet2

Table 4.1: Numbering convention for domain/site identifiers in Phosphorus testbed

The VLAN numbering scheme follows a pattern based on an ordered combination of the two domain identifiers. That is to say, two linked domains, if connected via tagged L2VPN, will generate a VLAN identifier constructed from a numeric prefix plus a combination of the numeric identifiers of each one of the domains. If several VLANs link two domains, a 1-digit prefix will be added to avoid VLAN identifier duplication.

Let X be the decimal, 1-digit, convened prefix of the testbed; Y, Z the numeric identifiers for two different domains and n the 1-digit prefix for duplication avoidance (n valued between zero and nine, both included). Then, VLANs between these domains will be tagged as either nXYZ or nXZY. It is important to highlight that ordering of domain identifiers matters. As a consequence, the maximum number of VLAN identifiers between two domains is 20, given a fixed prefix X.

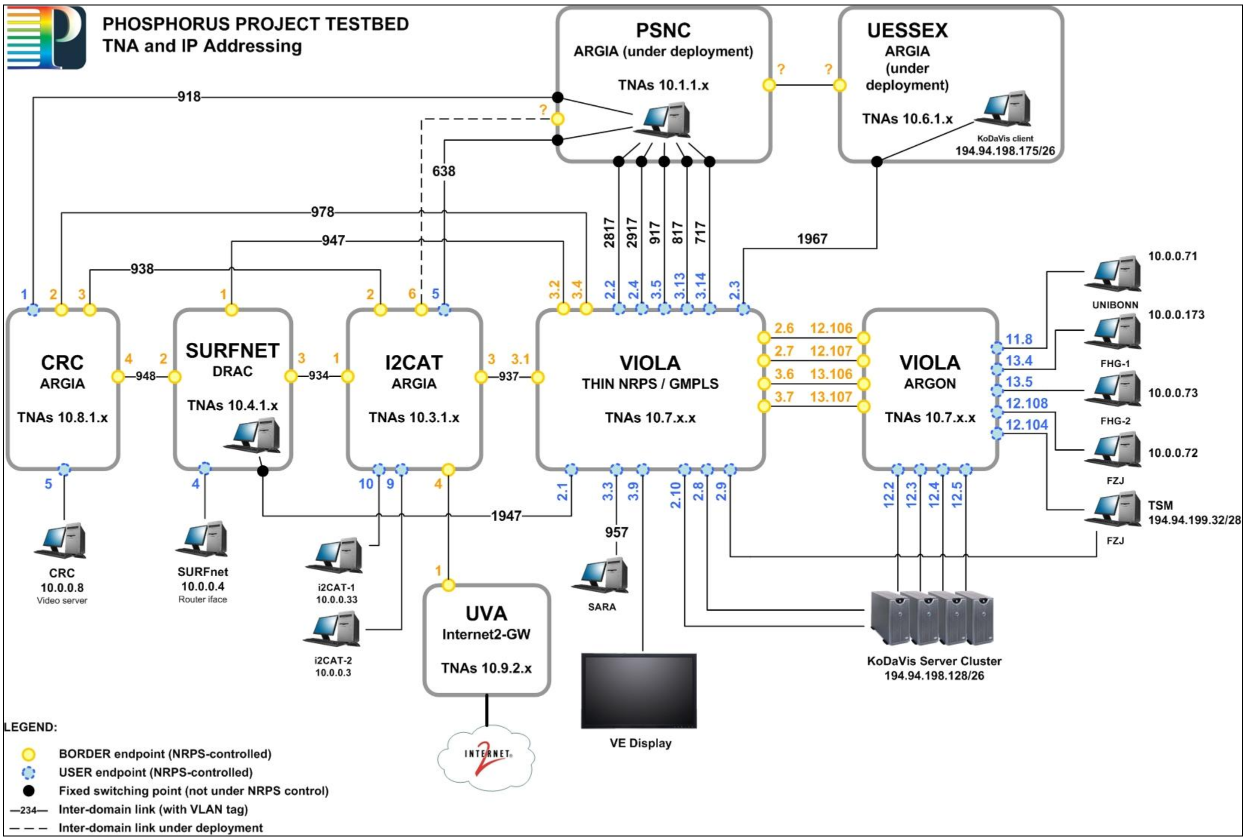

Figure 4.1: Data plane overview

Currently, the VLANs configured in WP1 testbed are:

VLANs IDs	CRC	I2CAT	SURFnet	VIOLA
VIOLA	978	937	947	
SURFnet	948	934		
I2CAT	938			
CRC				

Table 4.2: VLAN identifiers used in the testbed provided by WP1 members

Furthermore, other VLANs have been configured for connecting remote clients located in other partners' premises, such as PSNC, University of Essex or SARA:

- PSNC (ID=1) to VIOLA-GMPLS (ID=7): VLANs 717, 817, 917, 2817, 2917
- UESSEX (ID=6) to VIOLA-GMPLS (ID=7): VLAN 1967
- SARA (ID=5) to VIOLA-GMPLS (ID=7): VLAN 957
- Internet2 (ID=9) to I2CAT (ID=3=: VLAN 939

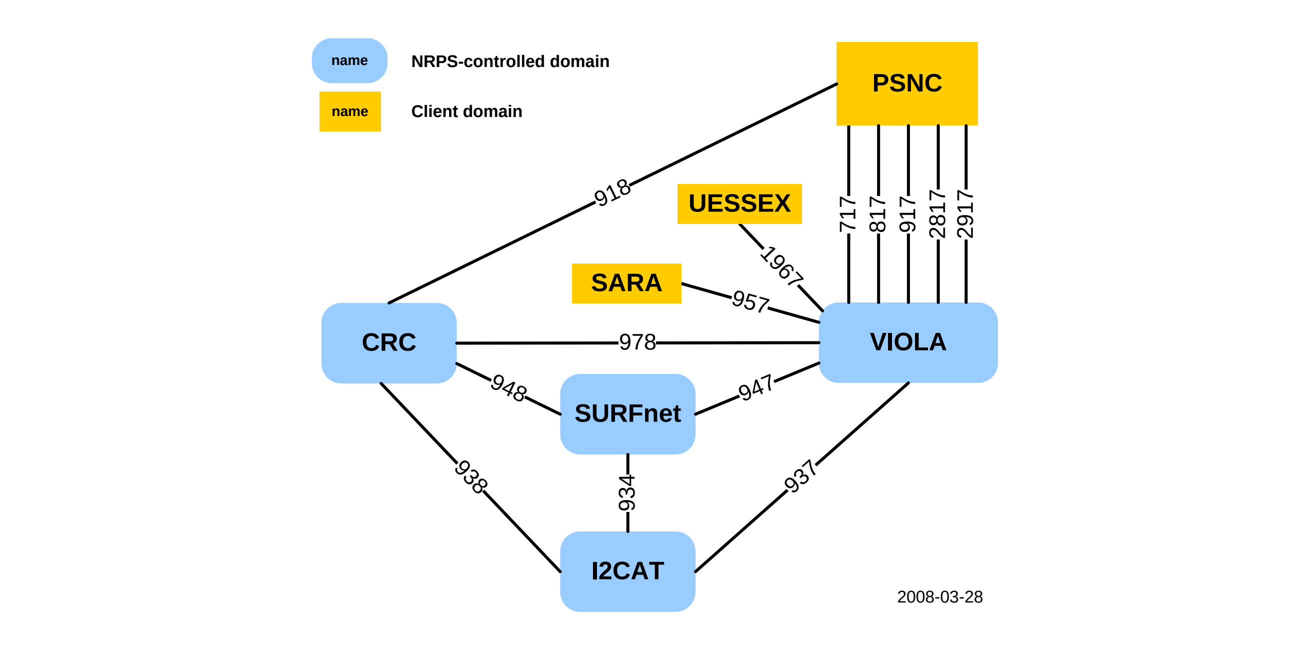

Figure 4.2: VLAN map and addressing.

4.1.2 Data plane addressing scheme

WP1, in conjunction with WP6, has followed own addressing schemes for the testbed as described in this section. As WP1 software prototypes deal with layer 2 resource provisioning systems, an addressing scheme has been proposed and convened among all partners to identify endpoints uniquely within the testbed. This addressing scheme is based on the numeric domain identifiers exposed before and follows an IPv4-like pattern, that is, every domain has its own TNA space with its own mask.

The endpoints belonging to the different domains are identified as follows:

Domain/Site numeric identifier	Domain Name	TNA space (address / mask)	Subspaces used	
1	PSNC	*Does not apply*	*Does not apply*	
3	I2CAT	10.3.1.0 / 24	UCLP	10.3.1.0 / 24
4	SURFnet	10.4.1.0 / 24	DRAC	10.4.1.0 / 24
5	SARA	*Does not apply*	*Does not apply*	
6	UESSEX	*Does not apply*	*Does not apply*	
7	VIOLA	10.7.0.0 / 16	GMPLS	10.7.0.0 / 21
			ARGON	10.7.8.0 / 21 10.7.128.0 / 21
8	CRC	10.8.1.0 / 24	UCLP	10.8.1.0 / 24
9	Internet2	10.9.2.0/24	IDC/DC	10.9.2.0/24

Table 4.3: Phosphorus TNA addressing scheme.

With respect to IP addressing, WP1 partners have defined an own IP addressing for test hosts based on private IP addresses within the range of the network address 10.0.0.0 with mask 255.255.255.0.

The tests hosts used regularly are compiled in the following list:

- CRC: 10.0.0.8
- SURFnet: 10.0.0.4
- I2CAT: 10.0.0.3, 10.0.0.33
- UniBonn: 10.0.0.71
- FHG: 10.0.0.73
- FZJ: 10.0.0.72
- Internet2: 10.0.0.91

In **Figure 4.1**, a full map of the Phosphorus testbed can be found, in which VLAN tagging, TNA naming and IP addressing are shown.

4.2 Interdomain control plane configuration

Signalling between the domains participating in the WP1 testbed consists of web service calls between the NRPS Adapters and the central IDB instance. As described in detail in [Phosporus-D1.4], the NRPS Adapters register by calling the addOrEditDomain operations of the higher IDB instance's Topology Web Service (Topology-WS), and the central IDB reserves resources in the different domains by calling the isAvailable and createReservation operations of the corresponding NRPS Adapters' Reservation Web Service (Reservation-WS).

A web service is identified by an *Endpoint Reference* (EPR) that contains the host name or IP address of the server running the web service. In the WP1 testbed, the control plane is not coupled with the data plane. Instead, the regular Internet connectivity is used for signalling between the different systems. To secure the testbed against unauthorized access from the Internet, a VPN has been set up based on the tinc software [tinc], following the recommendations of WP6 (cf. [Phosphorus-D6.1]).

The address scheme proposed in by WP6 has been adopted: The first octet of the IPv4 address is set to the decimal value 10, indicating that this is a private IP address (cf. [RFC1918]). The second octet is set to 1, indicating that this is a WP1 testbed address. The third octet is set to the number associated with the project partner. The fourth octet is assigned to different systems by the project partner hosting these systems.

Table 4.4 shows all control plane addresses currently in use in the WP1 testbed. The central NSP instance is maintained and hosted by the University of Bonn, therefore its VPN IP address is located in the VIOLA subnet.

Local testbed	Domain name	VPN IP address
I2cat	`i2CAT`	10.1.3.100
Surfnet	`surfnet-testbed`	10.1.4.1
VIOLA	*(central IDB instance)*	10.1.7.1
	`viola-mpls`	10.1.7.2
	`viola-gmpls`	10.1.7.3
CRC	`CRC`	10.1.8.1

Table 4.4: Control plane addresses within the WP1 testbed.

4.3 GAAA-TK Integration

As described in Sections 4.1 and 4.2 the WP1 testbed is composed of different domains following a specified numbering scheme and rudimentary security measures are already implemented. The latter and the more complex AuthN/AuthZ scenarios are described in the subsequent sections.

4.3.1 Level of Security

Security in this context can be located on the network, transport, and message level (cf. Figure 4.3). Within the WP1 testbed, the control plane communication is secured by using **network level security** (NLS). Each involved system is part of a virtual private network (VPN) that was created by using the free software tinc[tinc] (cf. Section 4.2). Communication from other locations (e.g. for the user GUI) is allowed for a reduced set of source IP address ranges only. Other systems could easily be added to the VPN or to the allowed address range.

Since the security within the Service Plane is based on NLS, no **transport level security** (TLS) mechanisms are implemented. In order to communicate with other systems it is necessary to provide security for this level using SSL, but for the specific gateway only. This was exemplarily implemented within the scope of the Internet2 IDC Gateway/Translator.

In order to integrate the GAAA-TK into the WP1 Service Plane **message level security** (MLS) mechanisms were deployed. Since the GAAA-TK does not handle AuthN issues but is based on successful authentication, the following discussion is parted in the authentication (AuthN) and authorization (AuthZ) phase.

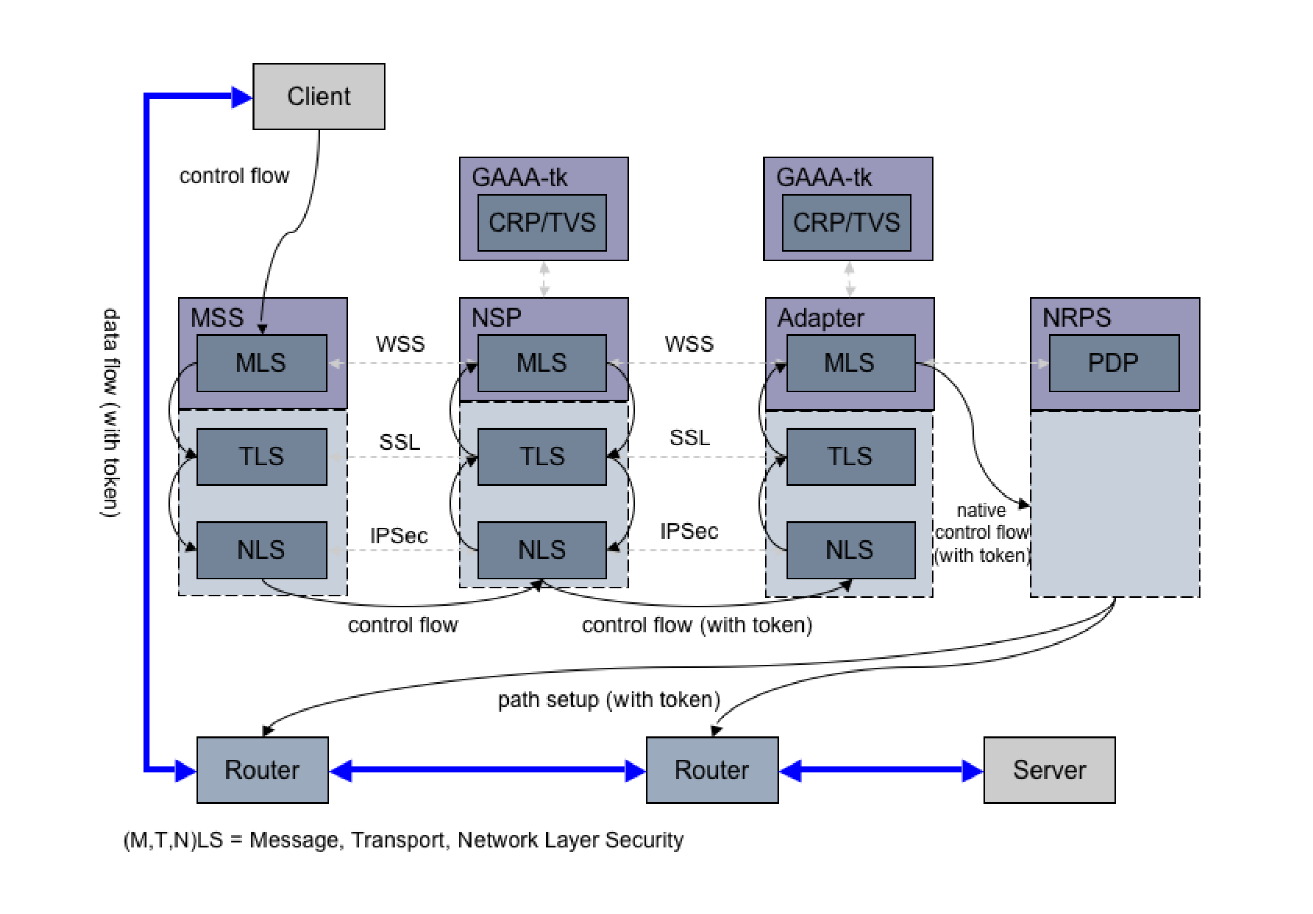

Figure 4.3: Overview of the security levels.

4.3.2 Authentication

The Harmony Service Interface contains a central module that is used for **AuthN** aspects. It is used to authenticate/decrypt all incoming and to sign/encrypt all outgoing traffic. This service is based on the OASIS Web services Security standard [WSS]. Besides procedures to sign and to encrypt SOAP messages the standard includes options to attach security credentials like username/password, X.509 certificates or tokens.

Figure 4.4 depicts a sequence of interactions needed for the authentication flow between interacting systems. The following sequence description is simplified and reduced to a single MSS (WP3 MetaScheduler), Harmony (WP1 Service Plane), and G^{2}MPLS (WP2 Service Plane) communication for AuthN aspects: (1) An MSS client sends a request to the Meta-Scheduling Service (MSS) with local user credentials. (2) The MSS authenticates and authorizes the user and the request locally. In case the request is authorized successfully, the scheduler maps the user credentials to accordant global attributes, adds these to the request for the NSP/IDB and signs the message with its private key. (3) The message then is sent to the NSP/IDB on behalf of the client. Since the public key of the MSS is trusted within the IDB, the message is accepted in the next step. Furthermore a complex authorization process has to be implemented in order to validate the request. Finally the signature of the valid incoming request will be removed and the request may be split into several new requests. (4) The outgoing messages to the G^2MPLS gateway are signed by the NSP/IDB. All authorization related information that may be added by the MSS is forwarded without any modification. (5) In the expected case that the G^2MPLS gateway trusts the NSP/IDB key all authorization information (e.g. global attributes, tickets) and the request is forwarded to the specific NRPS. (6) A complex authorization process has to be implemented in the G^2MPLS gateway or the underlying systems itself.

This way, the service plane acts as a transparent broker between the MSS and the G^2MPLS gateway. It is self-evident that this message level security flow is also applied for the corresponding response messages. Additionally, this architecture could be used to encrypt the whole message flow.

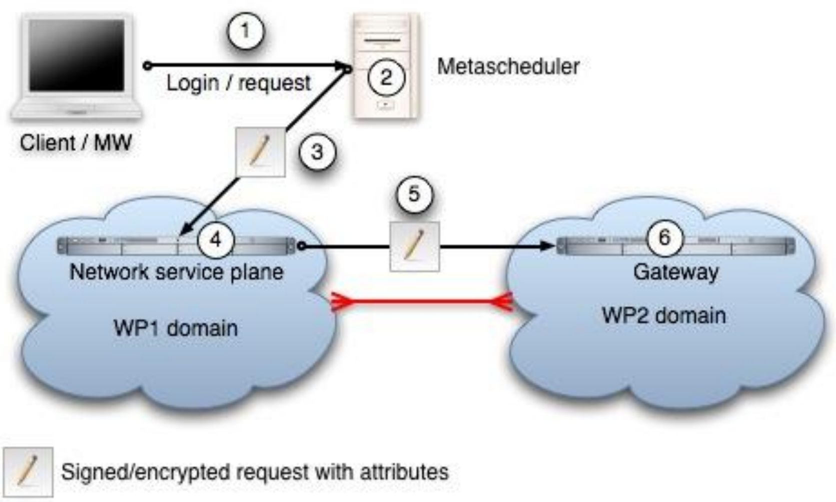

Figure 4.4: Authenticated message flow between MSS, NSP and G^2MPLS

4.3.3 Authorization

For the request **AuthZ** process WP1 is integrating the WP4s GAAA-TK into the Harmony Service Interface and will use it for all AuthZ related issues within the communication flow. As depicted in Figure 4.4 the MSS acts as an Attribute Authority (AA) that serves the role of a trusted entity for the service plane that mediates requests for holders of digital credentials. It must have privileged access to the local authentication domain database that holds information (identity attributes) about the credential holders. The MSS operates on rulesets defining what attributes can be attached to the request and under what circumstances. The service plane itself uses the GAAA-TK to authorize the request with its attached attributes. Then the NSP/IDB forwards the request with the user credentials (attributes) that are contained in the incoming message from the middleware to the involved domains. It is assumed that each domain has its own policy and attribute database and they may map the global attributes to local ones. In the case that global and local attributes are identical, this mapping reduces to the identity function.

Within the GAAA-TK, the path creation and path administration (and additionally its usage) is treated in different ways. This is why the two subsequent chapters describe the desired AuthN and AuthZ workflow in a more detailed way.

4.3.3.1 *Path creation*

In **Figure 4.5** a generalized AuthN/AuthZ workflow for a path creation process is depicted. The different steps can shortly be described as follows: (1) The client – in this case for example the MSS – creates [action| a reservation from A to B for a specific time frame [resource|. The request is signed by the clients certificate [credentials] and encrypted with the NSP/IDB/IDC/G^{2}MPLS-GWs public key. (2) The NSP/IDB/IDC/G^{2}MPLS-GW validates the signature of the incoming message by comparing it with the pre-installed public keys. (3) After the successful authentication parts of the request will be sent to an AuthZ module by using the GAAA-TK. This message includes a global reservation identifier (GRI) created by the NSP/IDB/IDC/G^{2}MPLS-GW, the action, resources and credentials. The AuthZ server in return will send back a token. (4) Now the NSP/IDB/IDC/G^{2}MPLS-GW creates a new reservation request for the involved NRPSs/NSPs/IDBs/IDCs/G^{2}MPLS-GWs including the token and the GRI. (5)(6) The next system now runs the AuthN and AuthZ process again for the incoming request. (7) Thereafter the underlying network elements (NEs) are configured (in case of an immediate reservation) and a local reservation ID (LRI) is sent back in step (8). (9) Finally the client receives the NSPs/IDCs/IDBs/G^{2}MPLS-GWs LRI, the GRI and the token for the reservation.

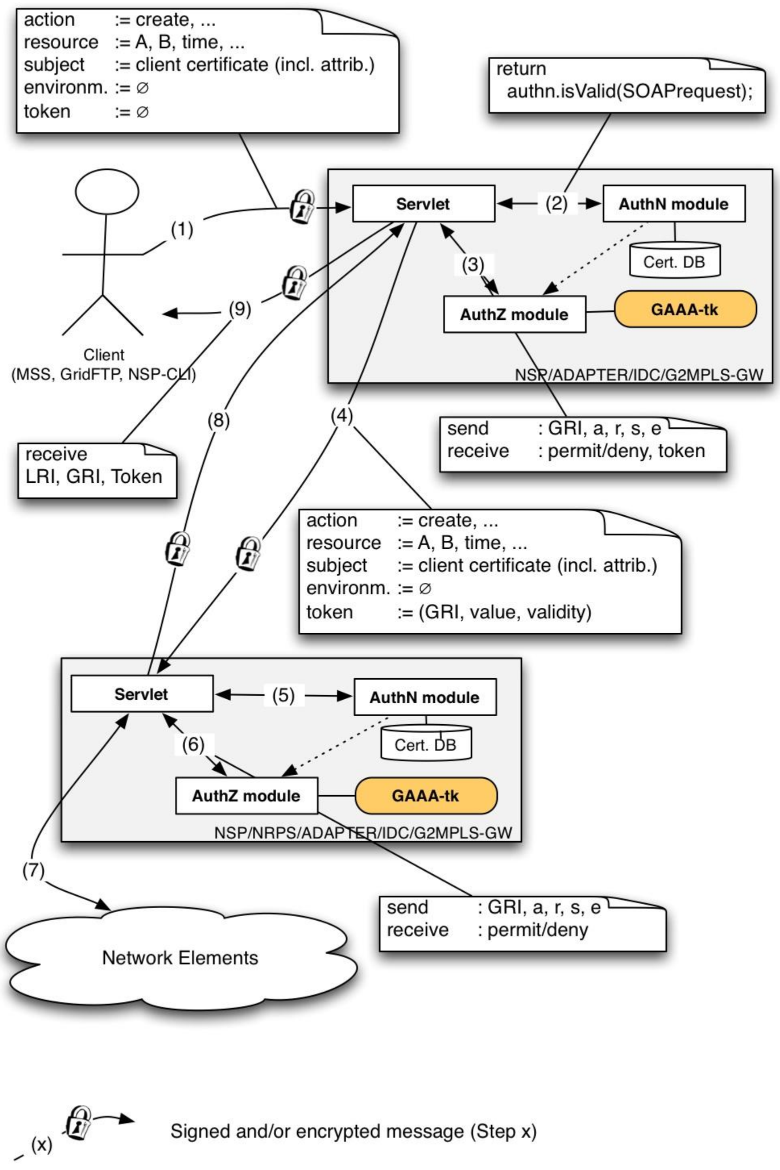

Figure 4.5: Generalized AuthN/AuthZ workflow for path creation

4.3.3.2 *Path administration*

In **Figure 4.6** a generalized AuthN/AuthZ workflow for a path administration process is depicted. The different steps can shortly be described as follows: (1) The client – in this case for example the MSS – wants to activate [action| a reservation identified by the GRI in the [token]. The request is signed by the client's certificate [credentials] and encrypted with the NSPs/IDBs/IDCs/G^{2}MPLS-GWs public key. (2) The NSP/

IDBs/IDC/G^{2}MPLS-GW validates the signature of the incoming message by comparing it with the pre-installed public keys. (3) After the successful authentication, parts of the request will be sent to a Token Validation Service (TVS) by using the GAAA-TK. The information sent to the TVS consists of the token and the user's credentials. The TVS in return will send a boolean value. (4) Now the NSP/IDBs/IDC/G^{2}MPLS-GW creates a new activation request for the involved systems including the token. (5)(6) The next system now runs the AuthN and AuthZ process again for the incoming request. (7) Thereafter the underlying network elements (NEs) are configured and a confirmation is sent back to the NSP/IDBs/IDC/G^{2}MPLS-GW in step (8). (9) Finally the client receives the confirmation for the activation.

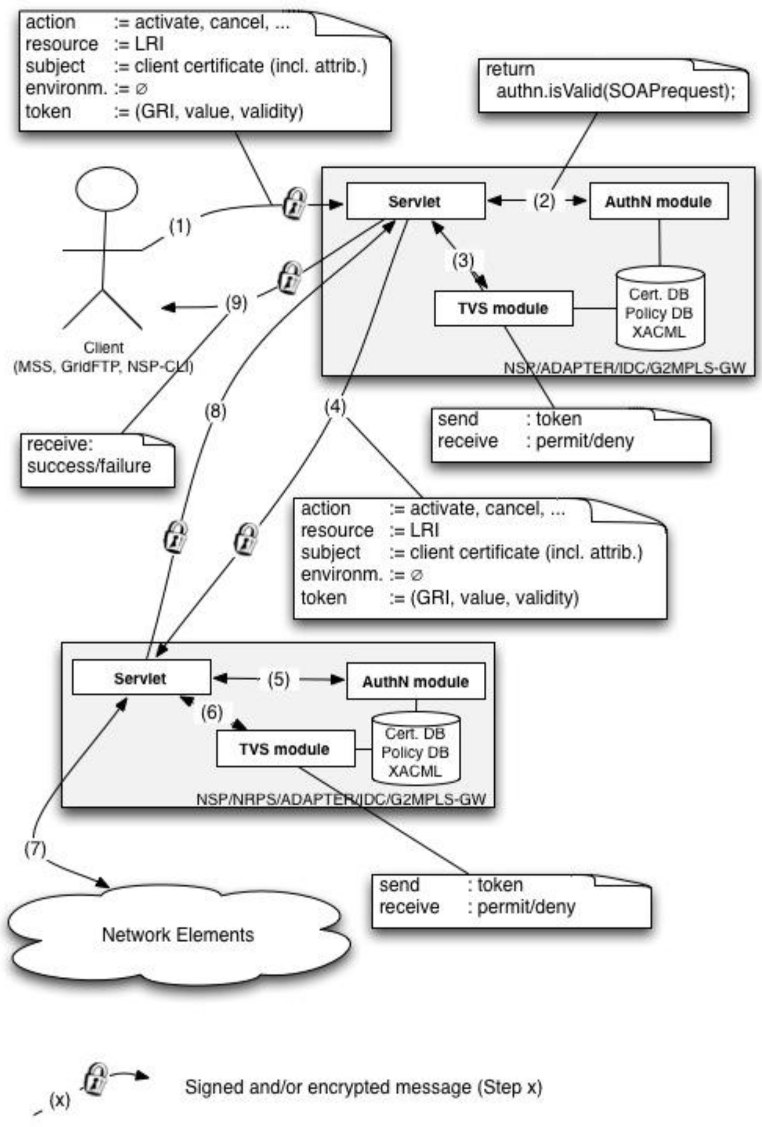

Figure 4.6: Generalized AuthN/AuthZ workflow for path administration

4.3.3.3 *Multidomain GAAA-TK Integration*

In **Figure 4.7** and **Figure 4.8** the current path creation and path administration scenarios for WP1 are depicted. They show how messages are forwarded between different domains and how the GAAA-TK is integrated in the different AuthZ stages. It's important to note that the GRI, in contrast to what is described in D4.2 Section 2.1, is not generated at the beginning. Instead it is generated after the underlying system has confirmed the reservation. This is why the AuthZ process is divided into two steps (3.5) and (9.5) in **Figure 4.7**.

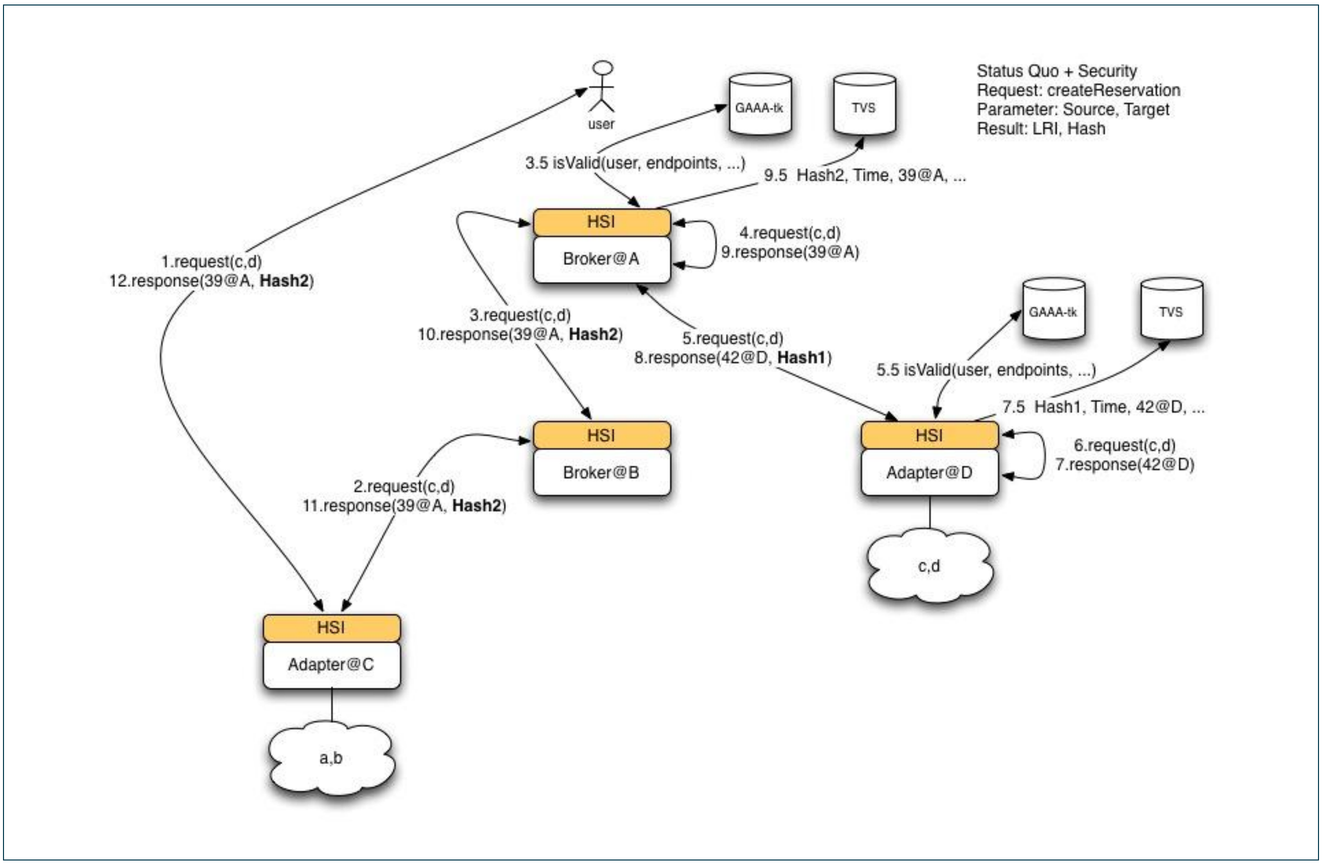

Figure 4.7: Multidomain GAAA-TK Integration Scenario (createReservation)

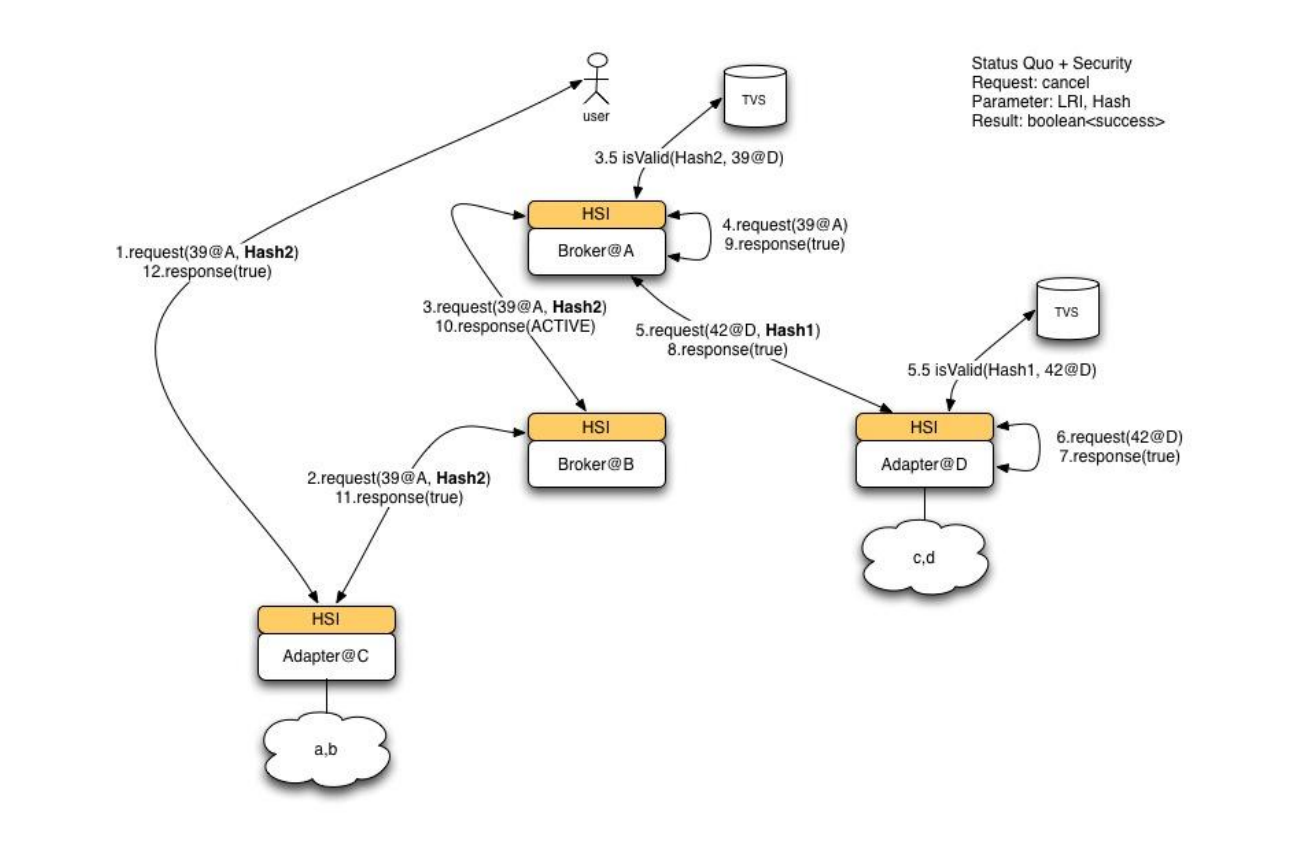

Figure 4.8: Multi-Domain GAAA-TK Integration Scenario (cancelReservation)

4.4 Use Cases and Test-bed Experiences

The current naming and addressing scheme of the WP1 Harmony testbed were successful transformed into a TNA based XACML-NRP policy profile. It permits reservations for a selected set of TNA address ranges (= corresponding domain) and actions for Phosphorus testbed users with specific roles. Thereby the following two scenarios are supported:

- Use Case 1: User/Group A is only allowed to use endpoints X, Y and Z
- Use Case 2: User/Group A is only allowed to use endpoints in domain N and M

Furthermore, after some iterations of the library, it was successful integrated into the Harmony Service Interface (HSI) and consequently part of each Harmony IDB, Adapter, and Translator. The informations that are needed by the GAAA-TK are extracted from the incoming request signature by the AuthN module (subjectId and subjectRole), the Harmony Request (action, endpoints, validityTime), and the Harmony Response (GRI). After filling the TVS table with the required information, the generated Token is send back to the user within the response message. The subsequent cancel request was authorized by matching the TVS table information with the given action, subjectId, subjectRole, GRI, Token. and validityTime.

Unfortunately with the current version of the GAAA-TK TVS each Token is only valid for a single action (e.g. cancel). This can be regarded as a requirement to the AuthZ decision integrity - according to this the authorisation and ticket is granted to what was requested. Extending this decision can be treated as a kind of delegation and typically entails either using special delegation policies or policy obligations. This requirement is crucial to use the GAAA-TK TVS within the WP1 testbed, since several actions are needed to administrate existing reservations. This leads us to another scenarios:

- Use Case 3: User/Group A is only allowed to invoke method X, Y, and Z
- Use Case 4: User/Group A is only allowed to invoke method X,Y, and Z based on session delegation

Finally all possible requests that are used within the Harmony interface (abbreviated in Use Case 3+4 as X, Y, and Z) must have a corresponding mapping within the XACML NRP policy description of each domain. After that and after solving minor technical issues, nothing should be in the way to fully integrate the GAAA-TK into the main WP1 testbed

5 Conclusions

The deliverable summarizes the results of the tests which were conducted as part of the PHOSPHORUS project by the end of September 2008. The tests aim to assess the ideas brought by the PHOSPHORUS consortium as well as verify the software developed by the consortium.

The ideas and developments of PHOSPHORUS are tested in a real optical network of European scope with a set of modern scientific applications which make use of the network and the PHOSPHORUS developments. This way the testbed emulates a modern GRID environment in which demanding applications running on computational nodes use a transmission network to exchange data between the nodes and access external devices.

This document shows the state of the tests as of September 30, 2008. Till this date some initial tests were done with the first outcome of other PHOSPHORUS activities.

The tests prove that the PHOSPHORUS ideas can be implemented in real networks and can benefit scientific applications giving them better control over the network and let them utilise the network in a more efficient way.

The tests made after September 30 will be reported in the next revision of this document, which will be submitted in the last month of the project duration.

6 References

EGEE	www.eu-egee.org
Globus	www.globus.org
IETF73	www.ietf.org
Internet2	www.internet2.edu
Tinc	www.tinc-vpn.org
www1	http://node01.PHOSPHORUS.man.poznan.pl/gridftp/
www2	http://node01.PHOSPHORUS.man.poznan.pl/tivoli_fzj/

7 Acronyms

AAA	Authentication, Authorisation and Accounting
AuthZ	Authorization
CP	Control Plane
EGEE	Enabling Grids for E-sciencE (European Grid Project)
FSC	Fiber Switching Capable
GAAA-AuthZ	Generic AAA Authorisation Framework
G.O-UNI	GRID Optical User to Network Interface
GÉANT2	Pan-European Gigabit Research Network
gLite	EGEE Grid middleware
GMPLS	Generalized MPLS (MultiProtocol Label Switching)
G^2MPLS	Grid-GMPLS (enhancements to GMPLS for Grid support)
I-NNI	Inter-Network to Network Interface
LSC	Lambda Switching Capable
LSP	Label Switched Path
NCP	Network Control Plane
NRPS	Network Resource Provisioning System
O-UNI	Optical User to Network Interface
ONL	Optical Networking Laboratory
Quagga	Software Routing Suite (www.quagga.net)
TBN	Token Based Networking
TOPS	Technology for Optical Pixel Streaming
TSM	Tivoli Storage Manager, a commercial backup/archive software by Tivoli, formerly known as IBM's ADSM)
UCLP	User Controlled Light Paths
UNI	User to Network Interface
VO	Virtual Organization
VOMS	Virtual Organization Membership Service

Disclaimer