034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number: D.5.5

Recommendations for Control Plane Design

Due date of deliverable: 31/03/2008
Actual submission date: 31/03/2008
Document code: Phosphorus-WP5-D5.5v1.0

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
Fundació i2CAT

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

Abstract

The first aim of this deliverable is to discuss some of the requirements imposed by the Grid applications to G^{2}MPLS Control Plane. Here, basically the three services of Advance Reservation, Advanced Routing and Advance Resources Scheduling have been taken into account. Then, on the basis of the G^{2}MPLS Control Plane architecture defined in the Phosphorus project, some required protocols extensions are highlighted.

Extensive simulation will be used to assess (for example in terms of scalability) the different functions offered by the Control Plane and to compare alternative interfaces and protocols. The simulation results will be the content of an Addendum to this Deliverable D5.5.

List of Contributors

George Markidis	AIT
Anna Tzanakaki	AIT
Kostas Katrinis	AIT
Christoph Barz	UniBonn
Markus Pilz	UniBonn
Kyriakos Vlachos	CTI
Emmanuel Varvarigos	CTI
Salvatore Spadaro	Fi2CAT-UPC

Table of Contents

List of Figures

0 Executive Summary

The interoperation of GMPLS-based networks with Grid infrastructures is not natively supported by the current GMPLS Control Plane. The design of integration strategies between Control Plane solutions in support of the Grid GMPLS Control Plane (G^2MPLS) have to address, among others, the design of the requirements imposed, by the Grid applications, to the Control Plane framework, in terms of protocols extensions and additional functionalities. Indeed, the high demanding Grid applications imposes strict functional requirements to the G^2MPLS Control Plane, which have to be fulfilled.

Grid middleware and applications couple distributed instruments, scientific data archives, and computing facilities with high-speed networks. Advanced optical networking is a promising candidate for addressing the emerging requirements of such high demanding applications. In this context, advanced routing is an essential functionality in G^2MPLS Control Plane, since it allows the integration of both Grid and network parameters into the end-to-end path computation process. In this process, issues such as optical constraints, connection types, multi-domain routing, resilience and coordination of both Grid and network resources can be addressed under the required advanced routing framework which has to be provided by a Grid-enabled Control Plane architecture.

On the other hand, resources scheduling decisions are usually taken by the central meta-scheduler or locally if a distributed meta-scheduler scheme is used. In any case, timing constraints, data and task workloads are communicated over the Control Plane and thus Control Plane protocols must be modified to carry specific fields for these parameters, apart from setting or tearing down bandwidth connections.

A common practice to cope with the complexity of scheduling network resources is to decouple the path selection decision, which is part of the admission decision, from its temporal aspects (it is discussed in Section 2.2). Within this Deliverable, Phosphorus also deals with the integration in the G^2MPLS of the advance reservation functionality.

Functional requirements

This Section deals with some of the requirements the high demanding Grid applications imposes to the G^{2}MPLS Control Plane. Specifically, firstly it concentrates on advanced routing, which includes issues such as optical constraints, resilience and multi-domain routing. Then, the advance resource scheduling and the advance reservations functionalities are discussed.

0.1 Advanced routing

Grid middleware and applications couple distributed instruments, scientific data archives, and computing facilities with high-speed networks, enabling new modes of scientific collaboration and discovery. Advanced optical networking is a promising candidate for addressing the emerging requirements of the high demanding applications. Advanced routing is an essential functionality in G^{2}MPLS Control Plane allowing the consideration of both Grid and network parameters into the path computation process. Issues like optical constraints, Grid application requirements, connection types, multi-domain routing, resilience and coordination of both Grid and network resources can be addressed under the advanced routing framework provided by a Grid-enabled Control Plane architecture.

In this section the functionality of advance routing and path computation that is required by the demanding Grid applications in order to provide end-to-end QoS connection provisioning is identified with respect to the G^{2}MPLS Control Plane proposed by Phosphorus. Architectures that motivate these routing approaches are also described.

0.1.1 Functional requirements for G^{2}MPLS

G^{2}MPLS Control Plane architecture is expected to expose interfaces specific for Grids and define a set of extensions to the standard ASON/GMPLS architecture resulting in a more powerful network Control Plane solution complying with the needs for enhanced network and Grid services required by network “power” users/applications (i.e. the Grids) [1]. In addition, the requirements of standard users that just need the

automatic setup and resiliency of their connections across the transport network are also supported by G^{2}MPLS. Functionalities related to the selection, co-allocation and maintenance of both Grid and network resources will be provided to configure network connections in the same tier of Grid resources, by guaranteeing service availability and tailoring to the user requirements.

A number of requirements that G^{2}MPLS Control Plane should support in order to provide advance routing functionalities are described below.

0.1.1.1 *Optical Constraints*

In all-optical networks the signals are transported end-to-end optically, without being converted to the electrical domain along their path. This reduces complexity and overheads and offers reduction of unnecessary and expensive optoelectronic conversions. However, due to the analogue nature of the optical networks as the optical signals propagate through the fibers, they experience several impairments degrading their performance. This has a direct impact on the dimensions that an all-optical network can support. To overcome the problems caused by the impairments at the physical layer, dynamic impairment management techniques may be implemented in-line (e.g. optical means of impairments compensation) or at the optical transponder interfaces (e.g. electronic mitigation of impairments). From the network layer view, the implementation of certain routing and wavelength assignments (RWA) algorithms that consider signal impairments and constraint the routing of wavelength channels according to the physical characteristics of the optical network paths can further improve the performance and minimize the blocking probability of connection requests. One approach that can be followed towards this direction is to identify a set of impairments that are considered independently and a path is assumed to be feasible if a set of criteria reflecting the signal quality in terms of the different impairments are satisfied. An alternative and more accurate approach is to identify a specific metric that considers all the physical impairments experienced by the optical signal and their interplay such as the performance factor Q as it is formed in the presence of a variety of impairments and consider this to be a routing parameter in the form of cost.

In [2] we extensively investigated these algorithms through a number of simulation studies, considering the Phosphorus network topology. Our impairment aware algorithms incorporated in their path computation process the most crucial linear (amplified spontaneous emission (ASE), polarization mode dispersion (PMD), crosstalk and chromatic dispersion) and non-linear (self-phase modulation (SPM), cross-phase (XPM) and four wave mixing (FWM)) optical impairments.

To overcome the limitations of the physical layer parameters through routing, the G^{2}MPLS Control Plane must be equipped with the appropriate impairment aware algorithms able to compute paths of adequate quality. To enable this functionality routing protocols (or signalling) should be extended to enable the dissemination of the optical parameters to the entities participating in the path computation. In addition, it would be of significant importance to empower the impairment aware algorithms with other functionalities and constraints like the ones described in subsequent sections (e.g., incorporate physical layer parameters for the protection path calculation or for the computation of the appropriate resource site when anycast routing is requested) leading to the identification of optimized paths through advance routing.

0.1.1.2 *Connection Types*

In grid networks typical point-to-point (p2p) connections might not be sufficient to satisfy the wide spectrum of Grid applications. For example, a particular end user or application may need to identify the availability of network and grid resources to access the required processing power or storage. From the network (routing) perspective this translates into the knowledge of the source point, while the destination is not determined and therefore intelligent mechanisms must be implemented to discover and identify the optimum end point and consequently the optimal path. Also in some cases (like the VLBI project) a number of grid users may simultaneously send a large amount of data to a single computation site forming a specific type of multipoint connection. Therefore, additional types of connections must be supported by the G^2MPLS Control Plane, in accordance to Grid user requirements as outlined here:

- **Point-to-multipoint**

  Point to multipoint is the connection type where single source of information is sending data to multiple locations, providing multiple paths. The special case of such situation is mentioned before as VLBI project, where data are sent in reverse direction from multiple sources to single destination point. The point-to-multipoint connection can be easily represented as network star topology model and represent an upcoming enhancement, e.g. for faster and effective data replication services.

- **Multicasting**

  Single-source Multicast is the communication mode that enables the delivery of identical data from a single source to multiple destinations simultaneously. The straightforward solution to this type of connection would be to establish one separate connection for each source-destination pair. However, this approach results in links carrying identical copies of data via multiple connections (wavelengths); which would clearly introduce waste of resources. Instead, multicast creates a spanning tree, with the source as root and the group receivers as leaves of the tree. Each switch that the tree comprises may have one or more children. In the case of a single child, the switch forwards the incoming data to the designated output port. When a tree node has multiple children, the switch replicates the incoming data of the multicast connection to multiple designated output ports.

  In order to realize optical multicasting various techniques should be applied both at the data plane and control plane. Data replication in the optical domain is achieved commonly through splitting of data arriving at an incoming port of an optical switch to multiple output ports. This is performed with full transparency and in many cases requires wavelength conversion capability at the branching nodes. On this basis, wavelength routed switches must be able to split a given incoming wavelength channel to multiple copies and switch each copy to a different output channel that may have the same or different wavelength compared to the original channel. Due to high manufacture and deployment costs and the limited multicasting requirements (at the wavelength level), a WDM network may only have part of the switching nodes equipped with light splitting and wavelength conversion capabilities (sparse splitting and sparse wavelength conversion).

It is the responsibility of multicast routing - implemented in the control plane - to compute the distribution tree, given the source and the receivers set and subsequently to provision all switches participating in the tree. In order to transfer data from one source node to multiple destination nodes in a WDM network, a multicast tree is normally required to be set up at the optical layer. Such tree-structured optical connections are known as light-trees. To establish a light tree, one needs to find a route from the source node to all the destination nodes and then assign available wavelengths to each link along the route.

- **Anycasting**

  Anycast is a type of data transmission where data are sent from single source to the nearest or best destination point as viewed by the routing topology. The group of receivers is identified with single routing address, however only one is receiving the data at the same time. The G^{2}MPLS implementation of such a service could be provided by processing at the Control Plane layer those Grid network services requests with implicit resource description (i.e. specification of just the involved Grid sites or the amount of Grid resources without any info on their network attachment points).

0.1.1.3 *Resilience*

The guarantee of the required service during the execution of a task might be compromised by some possible faults of the involved network or Grid resources. The Control Plane should provide means for faulty condition detection and reaction, as well as mechanisms for diverse routing between the working path and its backups. To deliver reliable services, G^{2}MPLS requires a set of procedures to provide protection or restoration of the data traffic in addition to advance routing for providing the protection or restoration paths. In case of failure occurrence on a working LSP, these recovery procedures have to be fast enough not to disrupt application's data connections and capable of performing rapid restoration even across multiple domains. The additional advantage of G^{2}MPLS for this situation stems from the ability of G^{2}MPLS to take care both of the network and grid resources and in combination with the anycast routing and the advanced reservation functionality, it can perform not only network recovery procedures but also initiate some grid recovery procedures without causing disruption to the application execution.

0.1.1.4 *Grid Quality of Service requirements*

Grid users require assurances that they will receive predictable, sustained and often high levels of performance from the network and Grid resources. Hence, network QoS provisioning is of great significance in Grids where performance characteristics that are of interest vary widely from application to application but usually include high throughput (bandwidth), low latency (delay and jitter) and low packet-loss rate (reliability). The resource selection and allocation algorithm for a Grid job cannot achieve the possible QoS objectives required by an application if its computations are completely unaware of the underlying network service. The required network service involves the network resources at the Grid locations and the interconnecting network domains. Various mechanisms exist to enable Grid network QoS. Admission control and scheduling, for example, can lead to altering the network topology by dynamically disabling or enabling link connections accordingly for satisfying

certain service requirements. Multi-constraint routing can also be used to provision feasible paths over which to route the appropriate tasks according to their requirements.

The QoS parameters that should be considered by such algorithms typically include:

- **Throughput**: An amount of grid data that can be transmitted between two locations in time unit (in grid terms)
- **Delay**: The time it takes for a packet to travel from the source (sender) to the destination (receiver). Large end-to-end delay between sites may result for some applications to not perform well (or at all).
- **Jitter**: The variation in the delay of packets taking the same route, prohibiting the support of many real-time applications.
- **Packet loss**: The rate at which packets are dropped, lost, or corrupted. Loss of a single packet often has a little impact on applications but repeated loss can have a significant effect.

0.1.1.5 *Inter-domain routing and scalability issues*

A significant issue of global communication networks is the difficulty to efficiently manage such networks. Indeed, large-scale networks are generally composed of smaller sub-networks, usually referred to as domains. The control and management of a single domain is performed locally, and information concerning state and availability is in general not shared with other domains. Special agreements (SLA - Service Level Agreement) are usually required between different domains to create peering connections and allow transit data transfers [2]. Interdomain and interworking of dynamic provisioning are two increasingly important requirements. Problems arise for the control and management of interconnections of domains, i.e. a multi-domain network, since their size and heterogeneity make it difficult to collect all information needed to make optimal management decisions. The scale of the network directly influences the number of events related to network state and availability; transferring this data to the controlling entities, and in turn processing it can generate a considerable overhead, leading to inefficient network operation.

As the network grows and support for services such as BoD is required, connection requests may arrive faster than the updating frequency of OSPF databases via the link state updates. Hence, decisions made by a path computation element (crossing multiple areas) will be outdated, possibly leading to computation of paths that are no longer available. Controlling the timing of when to send the state information, together with aggregation of this information (e.g. sending average values, aggregating information of multiple network links into a single value, use of abstracted information and so on), can significantly reduce the Control Plane overhead and improve the efficiency of bandwidth utilization. Towards this direction the Control Plane should address the trade-off between scalability and optimization of network resources.

0.1.1.6 *Coordination of Grid and Network resources*

Dynamic allocation of resources is clearly an important issue not just in allocating grid computing or storage resources, but also in allocating networking resources that are required to interconnect nodes and grids together. To allow co-allocation of Grid and network resources for provisioning optimized paths G^2MPLS must provide mechanisms for learning and advertisement of the Grid and network resource availability at the Grid

user site (Grid resource discovery). Also mechanisms for the negotiation of the Grid and network services configurable across the interface between the Grid user/site and the network are required (Grid service discovery). These discovery mechanisms will include both network specific resources and operating modes and Grid specific capabilities.

0.2 Advance resources scheduling

Scheduling decisions are usually taken by the central meta-scheduler or locally if a distributed meta-scheduler scheme is used. In any case, timing constraints, data and task workloads are communicated over the Control Plane and thus Control Plane must be modified to carry specific fields for these parameters, apart from setting or tearing down bandwidth connections. Functional requirements can be organized into two main categories; namely the fields that must be communicated to the grid resources and the fields that must be communicated to the central scheduler. Upon communicating these fields, the central meta-scheduler, will be able to allocate jobs to resources as well as reserve resources in advance.

User/Job fields: In the usual case, a User U generates tasks with probabilistic characteristics (workload, deadline, and requested number of CPUs) and probabilistic inter-arrival times. A task *i* requests to be executed on r CPUs (e.g. MPI parallel program) and has a computational workload L_i per CPU measured in millions instructions (MI), and a non-critical deadline $ti_{deadline}$, measured in seconds: by “non-critical” we mean that if the task’s deadline expires the task remains in the system until it is executed. Note that we have assumed that we know the exact workload of a task before its execution.

Grid resources fields: Each computational resource contains a number of CPUs, each with a number of processors with a computation capacity usually measured in millions instruction per second (MIPS). Every computational resource has a local queue where arriving tasks are stored, and a local queue scheduler which assigns the tasks of the local queue to the available CPUs. We have assumed that the local queue scheduler serves tasks in the FCFS order and can schedule the tasks to be executed in the future (in advance). In this way, a task that arrives at the queue and request to be executed at some specific time (which we will call Starting Time - t_{start}) reserves *r* CPUs for the duration that it requests (fixed advance reservation).

From the abovementioned features not all are directed to grid resources. Job delay is important when setting up network connections. If capacity is not available within the time constraints imposed by the task, job request must be discarded. In contrast task workloads, expressed in millions instructions (MI) or in storage capacity units are fields that are communicated only to computational resources. Every computational resource has a local queue where arriving tasks are stored, and a local queue scheduler which assigns the tasks of the local queue to the available CPUs. We have assumed that the local queue scheduler serves tasks in the FCFS order and can schedule the tasks to be executed in the future (in advance). In this way, a task that will arrive at the queue and request to be executed at some specific time reserves *r* CPUs for the duration that it requests (in the case of fixed advance reservation).

To this end, specific fields that the Control Plane needs to be communicated are a) timing constraints of the task execution and b) task workload. Figure 1 shows a potential Control Plane packet that is modified to include the abovementioned fields.

JOB ID	L_1	L_2	..	L_h	ST	TO	CPU/STO	D

Figure 0-1**:** The different fields of the setup packet for an optical grid Control Plane**.**

The routing path is specified as a sequence of link identifiers $L_1,L_2,\ldots,L_h$, Each node reads the first link identifier to determine the outgoing link to which it should be routed, and cyclically rotates the link identifiers so that the one just read becomes last. Basic fields of the control packet that must be communicated to all core nodes are:

- **Starting time (ST)**: Estimated starting time for job execution. The start time field ST specifies the time at which the reservation of grid resources for the specific task should begin. ST is relative to the arrival time of the job at the grid site. Therefore, the ST field is initially set equal to the round trip delay time and is updated at the intermediate nodes according to their resource availability.

- **Time offset (TO)**: It contains the time, following the reception of an acknowledge packet at the source, after which the source should start transmitting the data. The TO field is updated at every node in a way to be described later.

- **CPU (MI)/STO**: The computation and/or the storage capacity requested for processing/storing the job. It usually measured in millions instruction per second (MIPS) and GB respectively.

- **Task Deadline (D)**: it is the worst case delay tolerance of the task. The failure to be executed by that time D is considered as a deadline miss. If the deadline is not critical, then it may remain in the system to complete execution.

When node S_i receives the SETUP packet, it finds the first time t^i_{start} relative to the SETUP packet arrival that $t^i_{start} \geq ST_{i-1}$, and enough residual capacity is available to accommodate the transmission of the data of the job. We denote $\delta_i = t^i_{start} - ST_{i-1}$, where $\delta_i \geq 0$. If a t^i_{start} can be found, node S_i reserves the resources starting from time $t^i_{start} = ST_{i-1} + \delta_i$ and for time equal to the job duration. In that case of a successful reservation, node S_i updates the fields of the SETUP packet and forwards it to the next node. In particular, it updates the reservation starting time, time offset and reservation duration fields carried by the SETUP packet as follows: $ST_i = ST_{i-1} + \delta_i$, and $TO_i = TO_{i-1} + \delta_i$. It must be noted here that the above mentioned process refers to bandwidth resource, while grid resources are considered only at the last node. When the control packet arrives at the grid node, it again updates ST, TO values based on the CPU availability at the time. After proper reservation, the control packet has accumulated all the time offsets δ_i issued by the intermediate core nodes as well as destination grid nodes and the latter issue an ACK packet back to the source. The ACK packet contains the fields: $TO_{h-1} = \sum_{i=0}^{h-1} \delta_i$ Upon receiving the ACK packet, the source node, await for TO time and transmit the job.

0.3 Advance reservations

0.3.1 Advance Reservations in the scope of Meta-Scheduling

A set of demanding Grid applications are described in deliverable D5.4, which benefit of advance reservation in the scope of networking [15]. In the following, the functional requirements for future network service or Control Planes are derived on this basis. Two fundamental services are identified in the scope of networking. Firstly, inter-connections to stream or exchange information at high data rates are present in workflows of large scale Grid application. These advance reservations allow for synchronizing the work to be performed in a distributed environment, i.e. the network assists geographically distributed computational, visualization, and data streaming jobs. Secondly, the transfer of data for computational tasks is identified. The process of pre- and post-staging jobs has slightly different conditions. It is important to guarantee that the data – to be processed – is delivered timely. But neither the end-to-end delay of the connections nor the used capacity is crucial. It is assumed that these services can be scheduled beforehand with a dedicated guarantee (e.g. QoS, SLA) to support scheduling of workflows.

This section is structured as follows. In subsection 0.3.1.1 we present an architecture that motivates the usage of advance reservations in the scope of eScience applications with demanding network requirements. Subsection 0.3.1.2 briefly describes advance reservation types. Section 0.3.1.3 adds internal functional requirements, which are needed to provide the external functions independent of the network Control Plane architecture.

0.3.1.1 *Advance Reservations and Workflow Scheduling*

Research communities in areas like particle physics, numerical weather prediction, and bioinformatics use scientific applications with demanding needs for computational resources. Supercomputers, compute clusters, and mass storage systems provide a starting point to allow for these scientific applications (eScience). Consequently, scientific applications can effectively use a set of compute clusters and mass storage systems available in a computational Grid. If the workflow of such an application makes use of multiple resources, jobs can be distributed among resources in a parallel or sequential fashion.

In order to support applications with a need for coordinated usage of resources at different sites, advance reservations with guarantees (QoS, SLA) is beneficial. As local computational resource managers already provide support for coordinating a workflow of distributed applications by means of advance reservation (e.g. EASY, LSF, Maui, PBS Professional), the network has to be integrated in these architectures as well to support the execution of demanding applications in the Grid.

Following the findings in deliverable D5.4, we believe that few high demanding flows can be present which are either used for streaming/pipelining, MPI connections or delivery of large-sized data sets. If few sustained high capacity flows are using network resources, QoS or SLAs are needed, i.e. the objective is that booked resources are available when required.

Currently, meta-scheduling is an approach to integrate heterogeneous resources. Meta-scheduling aims at orchestrating resources from different organizations and locations. Here, we consider a meta-scheduler as an entity that is able to create a schedule that maps an application workflow onto available resources. This is performed by communication with a set of Resource Managers (RMs), i.e. resource management is done by local scheduling systems. For example, Compute Resource Managers (CRMs) are responsible for compute clusters, Storage Resource Managers (SRMs) provide functions for space management and data transfers.

In a basic setting, a meta-scheduler has to negotiate a timeframe for resource usage with multiple local schedulers. Three service types are identified in this setting:

1. The meta-scheduler has a specific start and end time that must be met by a service. For example, a common timeslot for computational resources are already identified and the meta-scheduler queries the network if communication constraints can be fulfilled for this time period.

2. The meta-scheduler has either a specific start or end time.

    a. The meta-scheduler has a start time for a service and requests a possible end time. For example, two jobs have a data dependency and the schedule includes a data transfer in between. In this setting a meta-scheduler has the end time of the first job and needs the duration for the data transfer to request resources for the subsequent job.

    b. The meta-scheduler has an end time for a service and requests a possible start time. In this setting the planning is performed backwards w.r.t. the former case.

In an advanced setting, the network can assist the process of meta-scheduling by at least two features: Resource preview and related requests.

0.3.1.2 *Types of Advance Reservations*

In accordance with deliverable D5.4 we assume a specific admission mode, i.e. advance reservations are guaranteed or rejected. It follows that a system creates a schedule for a set of resources to fulfil a reservation, i.e. a changing resource set may result in subsequent rejections. Subsequently, three types of advance reservations are quoted, which are presented in detail in D5.4. Furthermore, an extension to these types is described that supports the particular meta-scheduling scenarios from the previous section.

A basic form of an AR request is defined as follows and the life cycle is sketched in Figure 0-2: The request arrives at $t_{arrival}$, is admitted and starts at t_{start}. Furthermore, the usage phase (duration) is limited by t_{end}.

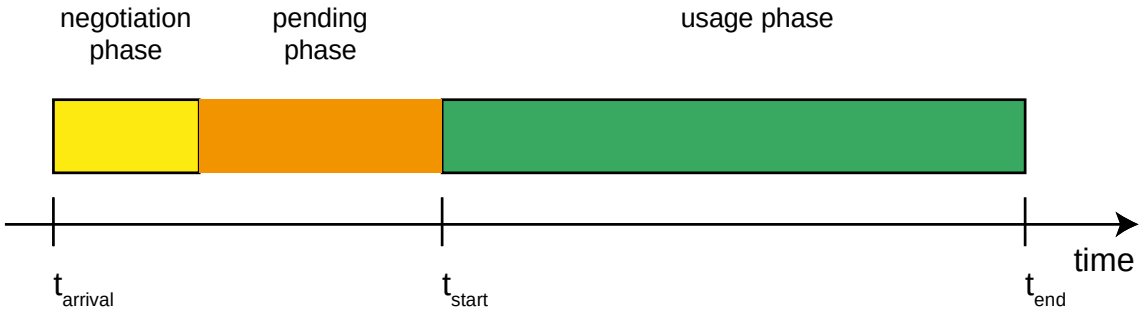

Figure 0-2: Arrival, start and end of a fixed advance reservation (FAR).

Fixed Advance Reservation Request: A fixed advance reservation request is defined as $FAR = (t_{start}, t_{end}, C)$ where $t_{start} < t_{end}$. The reservation starts at t_{start} and ends at t_{end}. The variable C represents additional resource constraints in the network domain, e.g. capacity.

The main idea of the so-called *deferrable advance reservations* (DAR) is to introduce a degree of freedom in the time domain, i.e. time related parameters define a range of possible values to establish the reservation. The life cycle of a deferrable advance reservation is given in Figure 0-3 and defined as follows.

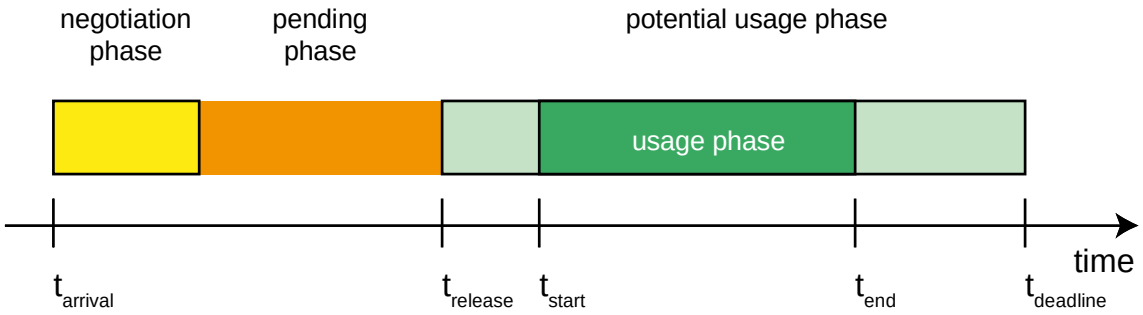

Figure 0-3: Timeline of a Deferrable Advance Reservation (DAR).

Deferrable Advance Reservation (DAR) Request: A deferrable advance reservation request is defined as $DAR = (t_{release}, t_{deadline}, d, C)$ where $t_{release} + d < t_{deadline}$. The reservation can start at $t_{release}$ and must end before $t_{deadline}$ the length of the usage phase is specified by duration $d > 0$. The variable C represents additional resource parameters.

Malleable advance reservations (MAR) are destined for data transfers. Hence, typical resource constraints for a MAR are a lower and an upper boundary for the transfer rate and a data amount to be transferred. The admission decision and the transfer information can be sent to the requestor at different points in time. The transfer information contains at least a starting and ending time.

Malleable Advance Reservation (MAR) Request: A malleable reservation request is defined as $MAR = (t_{release}, t_{deadline}, C)$ where $t_{release} < t_{deadline}$. Furthermore, the variable C represents additional resource constraints that can be used to determine possible durations of the reservation.

Generally, it is possible to adjust the transfer rate of a malleable advance reservation during the usage phase. This includes the possibility to interrupt a transmission for a while in order to find a new scheduling for the reservation request. It is up to the scheduling capabilities of the system in charge and the user capabilities, which strategies can be used.

In addition to these three types, two variations for each type are beneficial when workflows are scheduled. On the one hand, the start in known but the end time is to be determined. For example, tasks of a workflow are scheduled step-by-step starting from the first one. On the other hand, the deadline for a task is known in advance and the meta-scheduling system needs to know the latest start time to compute a schedule backwards.

0.3.1.3 *External Services*

The functional requirements described in this section are based on the aforementioned integration of network resources in the Grid environment consisting of a coordinating meta-scheduler that orchestrates heterogeneous resources. It is assumed that independent of the architecture of the network Control Plane, the requestor (meta-scheduler, Grid application) has a point of interaction to exchange information. Here, we concentrate on advance reservation specific functions, e.g. topology information may be exchanged beforehand.

The core functionalities can either be used to check the availability or to reserve resources. It is due to the system how these request “intentions” are implemented. In the scope of WP1 two different services are provided, i.e. availability requests are handled separately from reserve queries. In addition, complementing requests to cancel or modify a request are beyond the scope of this section.

Beside atomic requests which handle a single advance reservation in the network domain, an additional topic is the scheduling of multiple requests at once – for example a workflow represented by a directed acyclic graph.

1 Target Architectures

Two Control Plane models are defined for the G^{2}MPLS architecture, that is, G^{2}MPLS Overlay and G^{2}MPLS Integrated models [1].

These models concern the layering of grid and network resources. Thus, they have a different meaning and scope with respect to the IETF definitions for GMPLS Overlay, Augmented and Peer/Integrated models. A description of G^{2}MPLS Control Plane models is provided in the following subsections.

In the G^{2}MPLS Overlay model, the main scope of the Network Control Plane (NCP) is narrowed to Network Services and thus the service discovery and availability extensions for Grid sites are just flooded by routing and piggybacked by signalling at the different interfaces [3]. The Overlay model is intended to be deployed when most of the computational and service intelligence is maintained in the Grid layer which has Grid and network routing knowledge in order to provide Grid and network resource configuration and monitoring.

In the Phosphorus Integrated model, co-allocation services are directly implemented by the G^{2}MPLS NCP and Grid information (site capabilities and availabilities) is considered an integral part of the routing and the signalling decision processes. Therefore, in this case Grid extensions related to job resources (e.g. destination and characterization) are transparent at the different network interfaces, while other complementary information (e.g. related to data staging pre/post job execution) remains opaque and is simply piggybacked from the head-end to the tail-end of the network.

1.1 Advanced Routing

As stated in the architecture definition document [1], G^{2}MPLS adopts a hierarchical routing approach. Actually, this approach is deployed in case of interconnection of neighbouring domains through G^{2}MPLS NCP. On the other hand, the operation of a single G^{2}MPLS domain complies with the standard GMPLS requirement for a single routing area with flooding of TE information limited to that area.

The selected protocol for flooding TE information in a routing area is OSPFv2, with its standard extensions contributed by IETF for the GMPLS part and by OIF for the E-NNI routing part. G^{2}MPLS routing enhancements are conceived to be built on top of OSPF-TE and can be divided in three main classes:

- Extensions related to Grid resource characterization.
- Extensions related to scheduled resource availabilities, in order to support advance reservations for both Grid and Transport Network resources.
- Extensions for full-optical TE-link.

The extra traffic generated by Grid extensions in the GMPLS routing protocol is expected not to obstruct G^{2}MPLS node operations compared to the common low meshing degree of the experimental optical networks used for interconnecting Grid sites. Supporting studies and simulations are expected to assess the impact on routing traffic introduced by the proposed extensions in large scale networks. These studies will complement the experimental assessment of the G^{2}MPLS Control Plane tests that will be performed in some local test-beds of the Phosphorus infrastructure.

1.1.1 G^{2}MPLS Routing Controller (G^2-RC)

The G^{2}MPLS architectural breakdown in its main functional entities is described in details in [1]. The identified functional entities refer to equivalent components of the ASON architecture [4], but in most cases they provide the extended functionalities needed to support Grid Network Services. In this section the main entity responsible for routing is presented.

The G^{2}MPLS Routing Controller (G^2-RC) is the functional entity responsible for storing an updated topology view of Grid and network resources. The topology is detailed for the domain under its ownership and summarized at different extents for the neighbouring domains (i.e. just reachability information, reachability and inter-domain network connectivity, inter-domain and summarized intra-domain network connectivity; etc.).

The G^2-RC uses the topology for the computation of paths upon a request (with implicit or explicit declaration of the network attachment points). The computed path scope may range from portions of a route (at the minimum extend the next-hop) to the full end-to-end path across a chain of network domains (inter-domain routing). For this reason and for possible scalability issues, G^2-RC may be implemented either as a unique standalone module either as a distributed set of modules, in both cases complying with the IETF Path Computation Engine (PCE) architectural model [5].

The detail of the information stored in the G^2-RC topology and, consequently, of the computed routes depends on the adopted routing detail policy, i.e. the amount of information that each network operator configures and publishes internally (i.e. in its domain in case of distributed G^2-RC) and towards the neighbouring domains.

1.1.2 Connection types

The G^{2}MPLS call is an extension of the ASON call (an association between two or more users and one or more domains that supports an instance of a service through one or more domains) with further attributes, e.g. for temporal specifications.

The introduction of the Grid Network Service (GNS) transaction concept [1] allows fitting the typical Grid application use-cases in which multiple connections between different Grid sites (end-points) are requested for the execution of the unique job. Depending on the specific application, the network attachment point could be part or not of the calls composing the GNS transaction. For example, in case of distributed computing and visualization (Kodavis use-case) network attachment points need to be declared (explicit declaration), however in case of distributed storage they could not (implicit declaration).

In case of explicit declaration, the participating Grid sites are specified along with the respective network attachment points (i.e. TNAs). For this purpose, these information need to be available at the Grid layer and the resource allocation algorithm at this layer should adopt some mechanism for selecting the best match in a set of choices (i.e. the number of network attachment points for a Grid site could be more than one, e.g. for resiliency purposes).

In case of implicit declaration, two cases may occur:

- Participating Grid sites are specified, but the network attachment point is implicit, which implies that the G^{2}MPLS NCP will choose the best match in the set of the available network attachment points for the selected Grid site.

- Some of the participating Grid site are implicitly declared in the GNS transaction by asking for a service they can provide (e.g. an amount of CPU or storage), which implies that the G^{2}MPLS NCP will pick at first the best Grid site match in the set of the Grid sites for that application and its GNS transaction requirements, and then the best match for the network attachment in the set of those available for the selected Grid sites.

1.1.3 Optical and QoS constraints in G^{2}MPLS

With reference to the architectural and theoretical framework presented in [1] about optical impairments in all-optical networks, a number of additional parameters need to be added and flooded about all-optical TE-links for an enhanced constraint based path computation. For a G^{2}MPLS TE link belonging to an all-optical domain the following parameters should be specified and flooded via G^2.OSPF-TE:

- D_{PMD}
- Physical length (in Km)
- List of amplifiers (include their gain G and noise figure n_{SP})
- List of available wavelengths.
- Dispersion values
- Input power levels

Physical impairments should change slowly and relevant values/parameters could be measured at the installation time and possibly updated as a result of measurement campaigns. This relaxes the requirement for continuous and invasive measurements in the all-optical network.

The computation of a path request is driven by the PCE which in this case, represents a local Autonomous Domain (AD) that acts as a protocol listener to the intra-domain routing protocols, and is also responsible for inter-domain routing. PCEs peer across domains and exchange abstract or actual topology information to enable inter-domain path computation and also utilize G^2.OSPF-TE to share a link state database between domains. PCEs also include advanced algorithms which allow path computation with multiple constraints like physical layer impairments, network resource coordination, advance reservations and different connection types. The constraint path computation process performed by the PCE can be described briefly in the following steps. Prior to path request all necessary information on grid resources are identified and transferred to a MetaSheduling Service (MSS) responsible for the orchestration of resources of different sites belonging to different administrative domains, based on advanced reservations [6]. Also the Traffic Engineering Database (TED) is continuously updated with link state information driven from a routing daemon. Upon a request arrival the user specified parameters carried by the LSP request are parsed into constraints inside the PCE which takes the responsibility to compute the required end to end path if possible. This process can actually be quite complex and may require multiple interactions with MSS and PCEs in other domains. For example Grid applications may require the allocation of a number of different Grid resources for their execution in various scenarios. These resources usually are distributed on different Grid sites and dynamic optical network connection provisioning must be triggered considering all the resources required by the application. Connections may be initiated for example due to the necessity of code migration to other sites, for data retrieval and storage, for load balancing purposes, when parallel simulations are required or even for visualization of resources with real time characteristics. The coordination of such applications requires the rapid discovery of appropriate connections which are the result of very complex and intensive path computations. PCE can assist to this direction through the abstracted information of the global topology stored in the TED, of each domain, especially in cases where the network Management Plane is not able to provide this functionality.

The situation becomes even more complicated when Grid users request advance reservations to guarantee the availability of network resources. Advance reservation is a mechanism that allows a user to request exclusive access, for a specific time interval in the future, to a set of resources that satisfy specific requirements. Local Grid resources are basically under the control of Middleware's (MW) local schedulers of the Grid sites but the overall advance reservation of Grid and network resources is under the control of the G^2MPLS. Therefore, the reservations should be supported in two phases. In the first phase the MSS acting as a global Grid Broker must take into consideration the user time constraint demands and derive a Grid resource schedule by incorporating resources from different Grid sites. In the second phase the PCE of the G^2MPLS Control Plane must discover the available network resources according to the LSP schedule maintained at the PCE, necessary to provide access to the indicated Grid resources. In this way a reduced network topology is created which will be utilized by advance routing algorithms for optimal path computation. The LSP schedule information can be exchanged as resource reservation TE attributes in order to be translated into routing constraints understandable by the PCE. In addition the resource reservation states of the LSP schedule table at the PCE will be updated dynamically so that the path computation results can always be reflected in resource states and be disseminated to other domains. Therefore, any user requesting an advance schedule service can be provided instantly with a deterministic answer. Upon successful advance reservation, the reserved time slots of related resources will be dedicated to that service and will not be allocated to any other services. Also heuristics supporting malleable reservations can be introduced to the PCE, for cases where the job start time is not strictly fixed in order to avoid resource fragmentation that would lead to utilization degradation. The coexistence of immediate and advance reservations is another challenging task that algorithms implemented at the PCE

could deal with in an efficient way. In this case optimal resource partitioning (selected by the Phosphorus WP2) or pre-emption techniques can be implemented that will affect the routing decisions of the PCE.

Except from the Grid constraints (Grid resource scheduling and coordination) described above the PCE constructs the reduced topology of the network considering also QoS issues and policy restrictions. Network QoS provisioning is of great significance in Grid applications which require high throughput (bandwidth), low latency (delay and jitter) and low packet-loss rate (reliability) and can be deployed using various mechanisms. Admission control and scheduling for example can lead to altering the network topology by dynamically disabling or enabling link connections accordingly for satisfying certain service requirements.

Based on the created reduced topology the algorithms that are implemented on the PCE proceed to the path calculation taking into consideration several constraints related with the nature of the submitted job, the network and the available resources. One of the constraints that the PCE must handle deals with physical layer impairments. As signals propagate along the optical paths without OEO regeneration the physical impairments associated with the transmission line and the optical switching nodes accumulate resulting in some cases in unacceptable signal quality. Therefore, optical layer quality monitoring and optical quality–based network service provisioning (routing and resource allocation) become critical for connection SLA assurance. The implementation of certain RWA algorithms that consider signal impairments and constraint the routing of wavelength channels according to the physical characteristics of the optical network paths can improve the performance of connection requests. These algorithms are reported as Impairment Constraint Based Routing (ICBR) algorithms and ensure that connections are feasible to be established considering not only the network level conditions (connectivity, capacity availability etc) but also the equally important physical performance of the connections. The difficulties of a constrained based routing approach stem from the seemingly diverse nature of the networking and the physical performance issues that have to be considered. From the one hand, there are the physical layer impairments (linear like Amplifier induce noise, Polarization Mode Dispersion, Chromatic Dispersion, in-band crosstalk, filter concatenation and nonlinear like Self-Phase Modulation, Cross-Phase Modulation, Four Wave Mixing) that have to be taken into account and on the other hand there are the networking aspects (blocking probability, end-to-end delay, throughput) that capture and describe the overall performance of the optical network. It is evident, that all these heterogeneous issues have to be modelled and unified under a properly designed framework that will provide a solution for the RWA problem, which will be both feasible and efficient. For the realization of these algorithms by the PCE the majority of the impairments must be measured or derived using fast analytical modelling techniques on a link by link basis and therefore the PCE must interact closely with mechanisms that provide optical physical layer monitoring information. At the final step of the connection discovery impairment constraint based routing algorithms are applied and compute a path which is returned from the PCE in the form of an Explicit Route Object (ERO).

In conjunction with optical impairments, algorithms developed at the PCE should deal with resiliency strategies (protection, restoration) and provide solutions to support different connection types (unicast, multicast, anycast). Integrated approaches that consider multiple parameters can be developed to provide the ability to the PCE to compute optimal paths maintaining high network performance and implementing optimize protection and restoration schemes.

1.1.4 G²MPLS recovery

The delivery of reliable services is provided by G²MPLS through a set of procedures able to offer protection or restoration of data traffic. The recovery process can be distinguished according to area and time scale of the entire operation. Regarding area, the recovery procedure can be separated into:

- *Path-level* recovery procedures, in which failure notifications are propagated till the end nodes of the affected service and solved there.
- *Segment-level* recovery procedures, in which failures are notified and solved over a portion of the network traversed by a segment LSP.
- *Span-level* recovery procedures, in which failures are notified and solved locally at involved nodes, i.e. those next to the failed resource.

In terms of recovery time scale, faster and slower strategies can be distinguished. Fast recovery relies on pre-calculation and pre-allocation of a backup connection or a set of spans (i.e. protection) whereas slower strategies exist in which a new connection (or set of spans), that may be pre-calculated is dynamically allocated at the time of failure (i.e. restoration). As the recovery time scale can be performed at any recovery are (path, segment or span-level) an overall taxonomy of recovery may be sketched as in Figure 1-1.

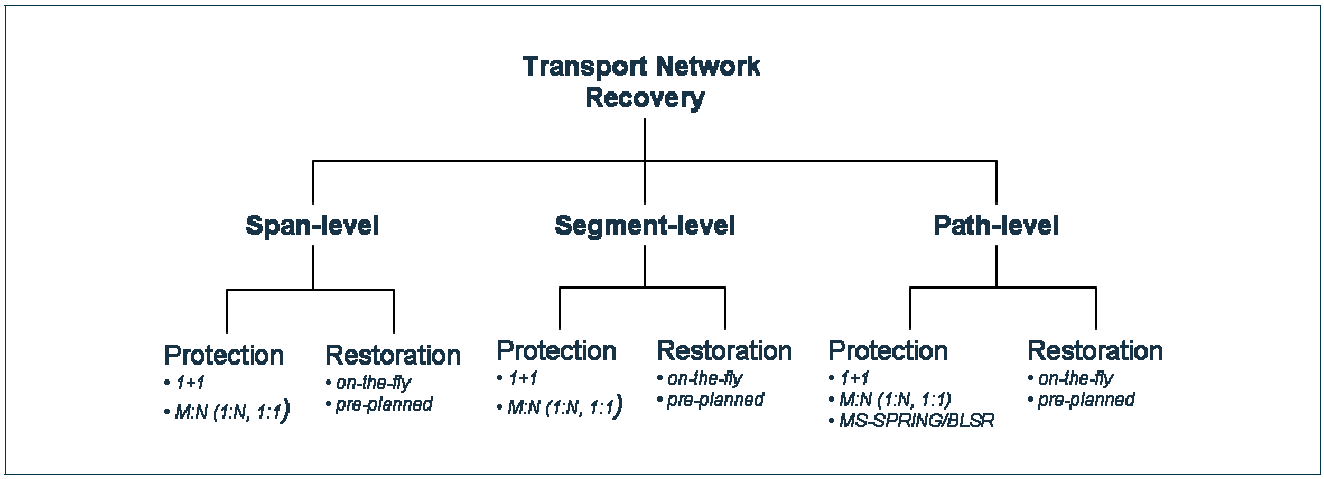

Figure 1-1: A taxonomy of recovery procedures.

Furthermore an important feature affecting any of the above restoration techniques is the centralized or distributed approach for the implementation of the recovery manager (path computation and signalling).

Centralized restoration techniques compute primary and backup resources in a central controller processing all the possible fault notifications generated in the domain under its control. This controller might be placed in the Control Plane, e.g. in the form of a PCE element [5] or in the Management Plane, i.e. in the Network Management System (NMS). In both cases, having the knowledge of the whole topology and the utilization of

the network at that time, the central controller can compute primary and/or backup paths and coordinate with the underlying network nodes by signalling (Control Plane case) or direct configurations (Management Plane case), in order to solve the faulty condition. Therefore, network intelligence is concentrated in a single point of the network and most nodes have limited Control Plane capabilities (e.g. only failure detection and notification capabilities).

On the other hand in distributed restoration that comprise the GMPLS and G^{2}MPLS preferred approach, every node in the network is empowered with full Control Plane capabilities and therefore can provide means for failure management and creation of backup paths.

The restoration time is mostly contributed by the time spent for failure detection, failure localization/isolation (if needed), for failure notification and also for the computation and allocation of the backup connection along the network.

1.1.4.1 *Protection and Restoration Procedures*

The existing GMPLS and G2MPLS support for protection procedures is briefly summarized below:

- *Dedicated 1+1*: In this case one dedicated protection LSP/span protects one working LSP/span and traffic is transmitted at the ingress node on both the working and the protection LSPs/spans not allowing any extra traffic to be transported over the protection LSP/span.

- *Dedicated 1:1*: One specific recovery LSP/span again protects one working LSP/span but the traffic is transmitted over only one LSP (either the working or the recovery). This enables the transportation of extra traffic on the unused LSP/span resources.

- *Shared*: One or more recovery LSPs/spans protect a number of working LSPs/spans all (working and recovery) ending on the same node pair. The working LSPs/spans is useful to be resource disjoint in the network, thus not sharing any failure probability

A summary of the GMPLS/G^{2}MPLS support for restoration procedures is also provided. Some further restoration schemes that are of interest in the G^{2}MPLS Control Plane architecture are obtained from the combination of the ones described below:

- *Pre-Planned*: An end-to-end restoration path is pre-calculated before failure and a signalling message is sent along this pre-selected path to reserve bandwidth. On failure detection, LSP signalling is performed along the restoration to actuate the cross-connections.

- *Shared-Mesh*: This is a case of Pre-planned restoration, where the pre-planning of protection LSPs can also include resources already planned for other protection LSPs.

- *On-the-fly*: Here an end-to-end restoration path is established after a failure occurs. This path may be dynamically calculated after the failure or pre-calculated before the failure and therefore no signalling is used along the path before failure and no restoration bandwidth is reserved.

In Phosphorus, where multi-technology and multi-domain schemes exist, multi-layer recovery management is aimed based on restorations. The existing hierarchies in Phosphorus are both horizontal (i.e. among similar technologies administratively partitioned) and vertical (i.e. among nested technologies).

Generally in a layered network, different technological layers may have their own different and distinct recovery mechanisms and recovery actions are taken by the recoverable LSPs/spans that are closest to the failure to avoid the multiplication and contention of recovery actions. Moreover, it is expected that a failure is properly correlated and isolated, in order to avoid notification on various deciding entities on different layers and consequently a race of concurrent recovery procedures.

The hierarchical restoration problem is faced in G^{2}MPLS according to four main directions:

- Restoration for inter domain network services, which include those means compliant with [7] for mitigating faulty network condition when inter-domain connections are involved.

- Coordination between the G^{2}MPLS layer and the Inter-domain layer (Network Service Plane and NRPS), which is needed when the network service is built through the interaction between G^{2}MPLS domains and e.g. NRPS-controlled domains capable of proper recovery strategies.

- Escalation of network recovery strategies, which makes possible to escalate a hierarchy of recovery procedures (e.g. protection, end-to-end restoration, etc.) in case of blocking in the adopted recovery action. In this case the user should accept the possible degraded network resiliency and, thus, this behaviour should be part of a proper agreement at the service discovery phase.

- Coordination among the different network layers, which includes those means needed to avoid race conditions between communicating layers, e.g. MPLS-TE and G^{2}MPLS.

1.1.4.2 *Joint network and grid restoration procedures*

For each job request in the Grid layer a number of network calls and connections are created in the G^{2}MPLS network and in case of fault in the network, each connection, segment or span could be recovered according to the procedures described above. However, some critical and unrecoverable fault condition might occur in the network, especially in case of limited connectivity and/or meshing degree. Depending on the seriousness and impact of the occurring network fault, it could be impossible for the NCP to recover the service. In such a case an escalation from the network layer to the Grid layer could be needed, triggered by timely fault notifications across the G.OUNI.

Fault notifications produced from the network should be received by those entities of the Grid middleware responsible for job monitoring and workflow check-pointing. These entities are expected to react at the notification e.g. by re-scheduling the failing/failed job in a different timeframe and/or on different Grid resources.

1.2 Advanced reservations

A common practice to cope with the complexity of scheduling network resources is to decouple the path selection decision which is part of the admission decision from its temporal aspects. A basic approach is to reduce the resource utilization information of every link in the considered time interval to a single value which is assumed to be representative for the whole interval. Treating the interval to have a uniform level of resource consumption allows for a simple application of a shortest path algorithm on the one hand, while neglecting the dependencies between neighbouring time slots via overlapping reservations and non-uniform resource utilization in the requested time interval. This approach may lead to a higher degree of resource fragmentation than an approach that has more insight into the level of resource consumption in the considered and in neighbouring time intervals.

1.2.1 Integration of Temporal Aspects into Routing Metrics

The main focus of this section is the extension of dynamic routing algorithms toward an online path selection strategy for a BoD service capable of advance reservation. For this purpose new routing algorithms or metrics respectively are introduced that explicitly take the temporal structure of the link utilization and the properties of the reservation request into account in order to reduce resource fragmentation. We present metrics for path computation that take into account the temporal aspects of a reservation. In addition, we define the following enhancements to previous approaches in temporal path selection strategies:

1. We introduce and classify various temporal metrics for link assessment.

2. We introduce an assessment of a reservation request to be served by a link by taking into account the change in temporal metric if a reservation is put on a specific link at the specified time.

3. The introduction of a time dependency in the routing metric, i.e. independent of utilization information certain time intervals are preferred, which is interesting for malleable reservations.

4. Different strategies of combining the above aspects in the path selection process for advance reservations.

1.2.1.1 *Definitions & Assumptions*

We assume that a timeslot-based resource management [8] is used to keep track of the available and reserved capacity for all links of the network. The timeline T is divided into a sequence of timeslots $Ti = [ti, ti+1[; 0 \leq i \leq b$

- *1* of dynamic length, where *[t0,tb[* represents the timeline and *Ti* *T* . Link utilization information is managed individually for each timeslot, i.e. accumulated values for every link are stored. Modelling the network topology as a directed graph *G = (V, E)* in which each edge *e* *E* has a non-negative link capacity *cmax(e) : E → N*. For each timeslot *Ti* we define the utilization (or reserved capacity*) u(Ti, e) : T x E →* *0* as the sum of capacities of all accepted reservations overlapping *Ti* on the link *e* and the residual capacity *r(Ti, e) : T x E →* *0 as r(Ti, e) = cmax(e) - u(Ti, e)*. When a new request is accepted the utilization and residual capacity values are updated. In general the time interval of a reservation may include several timeslots with different levels of resource utilization. We assume with no loss of generality that *t0* is the point in time at which a reservation request is received and *t0-tb* is the length of the book-ahead interval which is a sliding window in which new reservation requests can be accepted by the admission control mechanism (see Figure 1-2). For technical reasons, we define the utilization *u(Ti, e)* = 0 where *Tb = [tb, t∞[*.

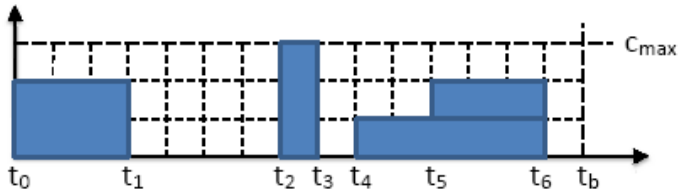

Figure 1-2: Timeslot-based resource management of network links.

An instantaneous allocation of residual capacities is called an immediate reservation (IR). Two basic forms of IRs can be differentiated: Either the duration of the resource usage is known in advance or not. An IR request with a known end time is seen as a special form of advanced reservations which is introduced below. The basic parameters of an IR request with unknown duration only contain the resource parameters. In the context of this deliverable, this includes the source of the requested path, the destination, and the required bandwidth (or capacity). Formally, a request for an immediate reservation is a tuple req = *(s, d, creq)*, where *s* and *d* denote source and destination respectively, *creq* denotes the required capacity. As an alternative to an immediate start time, the resource allocation can be in the future. This type of reservation is called advance reservation (AR). ARs are classified in two types that consider the relationship of time and resource constraints. The first type of AR can be related to the rigid tasks in the scope of scheduling and has independent time and resource constraints. Formally, this independent AR is defined as *reqiar= (tstart, tend, s, d, creq)*, where *tstart* specifies the start time of the request, *tend* is the end time. It is usually used to establish synchronous communication channels. The second type, a dependent AR, is specified by *reqdar= (trelease, tdeadline, s, d, datareq)*, where *trelease* specifies the earliest start time of the request, *tdeadline* the latest end time, and *datareq* a capacity-time product that specifies an amount of data that is to be transferred. It is used for asynchronous file transfers.

An algorithm that maps reservation requests to the available resources has to check for admission control whether *req* can be realized by the residual capacities in the network given by *cres*. If a new request is processed, the corresponding algorithm has to determine all timeslots which are overlapping with the request. Subsequently, the residual capacities can be inspected on all links within these timeslots and the algorithms are able to decide on the admission of the request. In a dynamic timeslot model [8], the computation of metric values for the path selection decision may require a temporal split of timeslots. If the request is feasible, the result is a set of reservation entities *R* *(T x T) x Ps,d x* $^{+}$ i.e. each reservation entity is determined by a set of coherent timeslots identified by their borders (*(T x T)*), a path from the source to the destination (*Ps,d*), and

a capacity value ( ). We assume that feasible IR requests and independent AR requests have exactly one reservation entity and there may be more than one reservation entity per dependent AR request.

1.2.1.2 *How to Integrate Temporal Information into Routing*

We define the cost function or *assessment function* of a routing metric as $m(e, c(e), u(T_i, e)) : E \times C \times U \rightarrow {}^{+}_{0}$. In addition, a concatenation operator is associated with every metric that defines how costs/assessment of single links are concatenated to the cost/assessment of a path *p*. Common concatenation operators are addition, multiplication, and minimum/maximum. The metrics are used by Ma and Steenkiste [9] can be interpreted as a general assessment of the link at the time of routing. This is reasonable due to their focus on IR. Although they used their routing algorithms to route bandwidth calls of known size through the network, they did not take information about the requested bandwidth into account. This is also true for the use of routing metrics to route AR calls, which in addition have a specified time interval. Burchard [10] took the full time interval of the request $[t_{start}, t_{end}[$ into account. Similar to the widest/shortest metric Burchard used the average utilization only as a discriminating additional metric in case of several minimum hop paths.

We define a general temporal metric which assesses the state of the link in the time interval $[t_{x1}, t_{x2}[$ as $m(e, c(e), u(T_i, e), t_{x1}, t_{x2})$. We define the cost or assessment function of a request dependent temporal metric as $m(e, c(e), u(T_i, e), t_{start}, t_{end}, c_{req})$. In the following we may omit parameters or use the index *i* instead of a time t_i if the meaning is clear.

1.2.1.3 *Classification of Temporal Link Metrics*

In this section, some general aspects to consider when developing temporal link metrics are discussed. Then, several metrics are presented, classified into two categories denoted Residual Capacity Metrics and Shaping Metrics, and their use in different circumstances and their advantages and disadvantages are presented in detail. There are some important aspects when dealing with temporal link metrics, namely passing of time, computational complexity, and a classification of the semantics of temporal link metrics.

1) Time: One aspect to consider is that time passes automatically. Between the admission control of two reservation requests, the book-ahead interval slides forward in time. This means that existing reservations might end and are removed, and new, unreserved capacity which has previously been outside the book-ahead interval now becomes accessible to new reservation requests. Assuming that the requested start times of the reservation requests are distributed over the whole book-ahead interval, this means that a reservation that begins far in the future is more costly than a reservation that is scheduled earlier in time. The reason for this is that the later reservation remains longer within the book-ahead interval and thus has a higher probability of blocking a new reservation request, while the earlier reservation leaves sooner and has a lower chance of interfering with future requests. A different way to see this is by considering the unused bandwidth instead. Unallocated bandwidth at the beginning of the book-ahead interval has a good chance of remaining unused, sliding out of the interval and thus going to waste. What follows is that ideally, a temporal routing metric should somehow reflect the temporal ordering of reservations.

2) Computational Complexity: Loadbased and interference-based routing metrics take the current load on the links in the network into account when making path selection decisions. Sophisticated path finding algorithms are used, which can become very complex and thus computationally expensive (see the MIRA algorithm [11] as an example). This is true even though the time domain is not taken into consideration. Having to calculate a temporal link metric value for each link in these path finding algorithms instead of using a simple link utilization value can increase the computational complexity considerably. A similar problem can arise when dealing with dependent ARs, trying to find a best-fit configuration and using a request dependent temporal metric. Here, for each configuration the temporal link metric value has to be calculated for each link in question, so again the overall computational complexity strongly depends on the complexity of the temporal link metric. It is thus very important that fast algorithms exist for computing the temporal link metric values. For all metrics presented here, only the points in time where the link utilization changes are important for the computation of the metric. As shown in [8], their number is bounded either by the number of accepted reservations *n* or the granularity of the system.

3) Classification: The temporal link metrics presented here can be divided into two classes:

Residual Capacity Metrics try to assess how much residual capacity a link has left for a given utilization timeline. They do not expect that any characteristics of future reservation requests are known, nor do they assume that certain reservation configurations are preferable.

Shaping Metrics on the other hand consider a link with a given utilization timeline as a geometric problem, with reservations represented by rectangles in a coordinate system with time on the x-axis and bandwidth on the y-axis (see Figure 4-2). Shaping metrics assume that certain configurations are preferable, either out of general considerations or because they expect reservation requests with certain characteristics to be more common than others. These metrics try to shape the overall geometric form of the reservations to match an optimal configuration as closely as possible. Without loss of generality, in the following all metrics are normalized to the interval $[0, c_{max}(e)]$ and expected to be maximized. If the metric values are to be used with the addition concatenation operator, the unreserved instead of the reserved capacity will be used, so that an unused link will have a lower weight than a fully utilized one.

A detailed list of temporal routing metrics can be found in [12].

1.2.1.4 *Application of temporal Routing Metrics*

The metrics described in the previous section might be seen as a general toolkit instead of a single strategy. In this section, we will discuss how this assessment can be used for the selection of paths for reservation requests, i.e. to construct reservation entities. Thus we present assorted samples of usage in the following sections.

Different Reservation Types: Temporal metrics can be evaluated for the whole book-ahead interval, but also for sub-intervals. For IR a reasonable choice is to take into account the whole book-ahead interval. As an alternative, the average usage duration of an IR can be used. Only residual capacity metrics can be used as the duration of the reservation is not known in advance. For independent ARs with a fixed start and end time, a

common subinterval for metric evaluation is $[t_{start}, t_{end}[$, which corresponds to the approach in [10]. However, for some metrics it might also be beneficial to include neighbouring timeslots. Both residual capacity metrics and shaping metrics can be used. In general, all temporal link metrics are also applicable to dependent ARs. Here, different sets of reservation entities can be compared, according to their metric values. This is equivalent to finding a best-fit configuration [10], [13]. In addition, the weight functions can be used to prefer configurations in a certain region of the book-ahead interval.

Application of the Link Assessment Metrics: Previous approaches did not use information about the requested bandwidth c_{req} for link assessment. A simple strategy to integrate this type of information is to temporarily add c_{req} to all links in the requested time period. This approach also incorporates the state of the link as it will be after this link is assigned to a path for the reservation with c_{req}. This might be of special interest for non-linear metrics/cost functions. Conceptually, one can identify a link assessment part and a reservation assessment part. The latter specifies only the change in state of the link assessment for the reservation. This differentiation is depicted in Figure 1-3 which shows a non-linear cost function of a link with u being the current link utilization corresponding to a metric value $m(u)$.

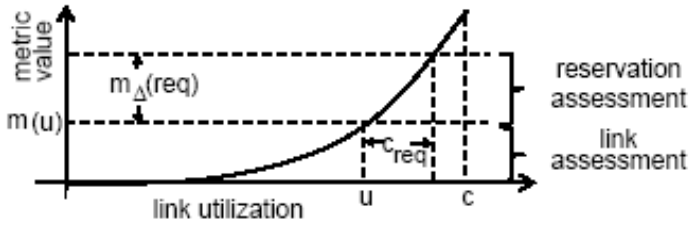

Figure 1-3: Metrics for link and reservation assessment.

A difference metric for a reservation request of capacity c_{req} is defined as

$$m_{\Delta}(req_{ar}) := m(u'(T_i, e)) - m(u(T_i, e))$$

where $u'(T_i, e) = u(T_i, e) + c_{req}$ if the reservation overlaps the timeslot T_i and $u'(T_i, e) = u(T_i, e)$ otherwise. While the residual capacity metrics can be used with and without reservation assessment, shaping metrics are ideal to be used as a difference metric concerning a request. In a dynamic timeslot model, timeslots might have to be split temporarily before the metric values can be computed. The metrics described here are robust concerning the split operation. Difference metrics can assess how well a particular reservation fits onto a link concerning the strategy of the resource management. In particular this is not limited to timeslots in the requested interval, but instead the whole book-ahead interval might be taken into account. This allows for an effective incremental computation strategy. As an example, the positive utilization gradients metric is an ideal candidate to be used as a difference metric. While the current state of fragmentation of a link might be of lower interest for the path selection process, the difference in metric values directly reflects how a reservation changes the state of fragmentation of a link. Therefore links should be preferred in the path selection process where the current reservation decreases the degree of fragmentation.

Combining Metrics

The metrics described in section IV reflect different aspects of the resource utilization of each link. Some of the metrics may even complement each other. In general, there are two different strategies to approach a path selection process with multiple criteria. The first one is to define a total assessment function that balances the different partial assessment functions in a single assessment value.

The widest/shortest path strategy uses the number of hops on the path as an optimization criterion which reflects resource consumption but does not take into account fragmentation. The widest criterion serves as a discriminator in case of several shortest paths and is traditionally used with the minimum residual capacity metric. However, also other link and reservation assessment metrics can be used as a discriminator. The concatenation operator for the discriminator and the optimization criterion do not need to be the same. Similarly, the metrics can be combined with the shortest distance metric. In general, one can define a weighted sum of different metrics or their reciprocals as long as it is guaranteed that the weighted sum for all links is not negative. This may be the case when using difference metrics. The second strategy is a search for a set of paths for every metric used. As an additional step one solution is selected.

1.2.2 Scheduling Advance Reservations

In the scope of G^{2}MPLS, availability information (calendars) is distributed via OSPF-TE extension to all nodes in a domain. Accordingly, all nodes in a network domain know the current and future state of network links. This information is – of course – restricted by additional parameters (e.g. number of timeslots to be stored per link).

In the following an overview on the mechanism to support advance reservations is presented.

1.2.2.1 *Fixed Advance Reservations*

The resource information of the time interval involved in the request is queried by advance reservation strategies to determine whether a request is feasible or not. The time complexity of the path selection (or path routing) is dependent on the constraint set. If only one connection with link constraints (e.g. minimum capacity) is requested, a shortest path algorithm with a polynomial running time identifies a candidate using a constrained topology. If multiple paths are requested at the same time or other path constraints (e.g. loss rate, delay) are included, the complexity of the path selection process can increase to a super-polynomial time complexity. In these cases, heuristics with potentially suboptimal results can be applied to keep the processing time low, e.g. multiple paths within a request are processed independently one after another.

1.2.2.2 *Deferrable Advance Reservations*

A deferrable advance reservation has a certain degree of freedom in the time domain. In particular, time related parameters define a range of possible values to establish the reservation. Compared to a fixed advance reservation, the parameters t_{start} and t_{end} can be determined by the system. Various strategies can be introduced to handle this kind of reservations. A straightforward approach is to scan the possible time intervals

given by the link availability information (calendars) and map the request to fixed advance reservations following a first fit approach.

1.2.2.3 *Malleable Advance Reservations*

A specification of the exact transmission rate can be omitted when a fixed amount of data has to be transmitted. Only general capabilities of the sender and the receiver, such as the maximal transfer rate and timing constraints for the transmission like a fixed deadline or an earliest starting time (release time), have to be regarded. By joining the time and resource constraints, the reservation system can find the most efficient solution for the requested transmission. This kind of reservation is denoted as malleable advance reservation [10] (or advance cumulative reservation [14]). A motivating example regarding efficiency is to fill gaps between allocated resources which are caused by accepted reservations. A detailed description of mapping malleable reservation requests to fixed advance reservations is included in D5.4 [15].

2 Final recommendations

WP5 investigates advanced Grid routing algorithms which could be integrated into G^{2}MPLS to provide QoS assurance, and reliable connection provisioning. Moreover WP5 aims to provide mechanisms for supporting alternative connection types and addressing inter-domain routing and the resulting scalability issues enabling coordination of grid and network resources. Specifically the next step in this WP is to perform some dimensioning studies on calendars information and depths to derive some a-priori evaluation of the routing and signalling load and investigate through large-scale simulation studies the scalability issues arising.

In the scope of G^{2}MPLS, availability information (calendars) is distributed via OSPF-TE extensions to all nodes in a domain. Accordingly, all nodes in a network domain know the current and future state of network links. This information is, of course, restricted by additional parameters (e.g. number of timeslots to be stored per link). The resource information of the time interval involved in the request is queried by advance reservation strategies to determine whether a request is feasible or not. The time complexity of the path selection (or path routing) is dependent on the constraint set. If only one connection with link constraints (e.g. minimum capacity) is requested, a shortest path algorithm with a polynomial running time identifies a candidate using a constrained topology. If multiple paths are requested at the same time or other path constraints (e.g. loss rate, delay) are included, the complexity of the path selection process can increase to a super-polynomial time complexity. In these cases, heuristics with potentially suboptimal results can be applied to keep the processing time low, e.g. multiple paths within a request are processed independently one after another.

Scheduling decisions are usually taken by the central meta-scheduler or locally if a distributed meta-scheduler scheme is used. In any case, timing constraints, data and task workloads are communicated over the Control Plane and thus Control Plane must be modified to carry specific fields for these parameters, apart from setting or tearing down bandwidth connections. Functional requirements can be organized into two main categories; namely the fields that must be communicated to the grid resources and the fields that must be communicated to the central scheduler. Upon communicating these fields, the central meta-scheduler, will be able to allocate jobs to resources as well as reserve resources in advance.

3 References

[1] PHOSPHORUS WP2, “The Grid-GMPLS Control Plane architecture”, deliverable D2.1.

[2] PHOSPHORUS WP5, “Grid Job Routing Algorithms”, Deliverable D.5.3.

[3] PHOSPHORUS WP2 “Routing and Signalling Extensions for the GMPLS Control Plane”, Deliverable D2.2.

[4] ITU-T G.8080/Y.1304 Recommendations, “Architecture for the Automatic Switched Optical Network (ASON)”, 2001.

[5] A. Farrel J.-P. Vasseur, J.Ash, “A Path Computation Element (PCE) – Based Architecture”, IETF RFC 4655, August 2006.

[6] Christoph Barz and Markus Pilz, “Co-Allocation of Compute and Network Resources in the VIOLA-Testbed”, TERENA 2006.

[7] T. Takeda, Y. Ikejiri, A. Farrel, J.P. Vasseur, “Analysis of Inter-domain Label Switched Path (LSP) Recovery”, IETF Draft, work in progress, draft-ietf-ccamp-inter-domain-recovery-analysis-00.txt, December 2006.

[8] C. Barz, U. Bornhauser, P. Martini, and M. Pilz, “Timeslot based resource management in grid environments,” in Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems, Innsbruck, Austria, February 2008.

[9] Q. Ma and P. Steenkiste, “On path selection for traffic with bandwidth guarantees,” in ICNP’97, 1997, pp. 191–202.

[10] L.-O. Burchard, “Advance reservations of bandwidth in computer networks,” Ph.D. dissertation, Technical University of Berlin, 2004.

[11] K. Kar and M. Kodialam, “Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, vol. 18, no. 12, pp. 2566–2579, December 2000.

[12] C. Barz, M.Pilz, A. Wichmann, “TEMPORAL ROUTING METRICS FOR NETWORKS WITH ADVANCE RESERVATIONS”, accepted at the Workshop on High Performance Grid Networks, co-located with CCGrid 2008, Lyon, France, May 2008

[13] C. Barz, P. Martini, M. Pilz, and F. Purnhagen, “Experiments on network services for the grid,” in Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN ’07), 2007, pp. 45–54.

[14] R. Guerin and A. Orda, “Networks with advance reservations: The routing perspective,” IEEE INFOCOM 2000, vol. 1, pp. 118–127, 2000.

[15] PHOSPHORUS WP5, “Support for Advance Reservations in Scheduling and Routing”, Deliverable D.5.4.

4 Acronyms

ASE	Amplified Spontaneous Emission
E-NNI	Exterior NNI
ERO	Explicit Route Object
FCFS	First Come First Serve
FWM	Four Wave Mixing
G.OSPF-TE	GMPLS OSPF-TE
GNS	Grid Network Service
G²MPLS	Grid-GMPLS (enhancements to GMPLS for Grid support)
GMPLS	Generalized MPLS
G-OUNI	Grid OUNI
IETF	Internet Engineering Task Force
LSP	Label Switched Path
MPLS	Multi Protocol Label Switching
NCP	Network Control Plane
NMS	Network Management System
NNI	Network to Network Interface
NRPS	Network Resource Provisioning Systems
OIF	Optical Internetworking Forum
OSPF	Open Shortest Path First protocol
OSPF-TE	OSPF with Traffic Engineering extensions
O-UNI	Optical UNI
PCE	Path Computation Element
PMD	Polarization Mode Dispersion
QoS	Quality of Service
RWA	Routing and Wavelength Assignment
SLA	Service Level Agreement
SPM	Self-Phase Modulation
TE	Traffic Engineering
UNI	User to Network Interface
XPM	Cross-Phase Modulation