034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number: D.2.2

Routing and Signalling Extensions for the Grid-GMPLS Control Plane

Due date of deliverable: 2007-09-30
Actual submission date: 2007-09-30
Document code: Phosphorus-WP2-D2.2

Start date of project:
October 1, 2006

Duration:
30 Months

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

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

Abstract

This document describes the Grid extensions for the GMPLS routing and signaling protocols, with specific protocol objects for G.OSPF-TE and G.RSVP-TE. These extensions are integral part of the G^{2}MPLS Control Plane prototype delivered by Phosphorus WP2. Most of the extensions are also part of the G.OUNI routing and signalling protocols, used to enable the integration with Grid middleware towards Grid Network Services. Architectural backgrounds for this deliverable are “The Grid-GMPLS Control Plane architecture” described in D2.1 and the “Grid-GMPLS network interfaces specification” delivered by D2.7.

Project:	Phosphorus
Deliverable Number:	D.2.2
Date of Issue:	30/09/07
EC Contract No.:	034115
Document Code:	Phosphorus-WP2-D2.2

List of Contributors

Nicola Ciulli	NXW
Gino Carrozzo	NXW
Francesco Salvestrini	NXW
Giodi Giorgi	NXW
Giacomo Bernini	NXW
Artur Binczewski	PSNC
Bartosz Belter	PSNC
Radoslaw Krzywania	PSNC
Damian Parniewicz	PSNC
Maciej Strozyk	PSNC
Dimitra Simeonidou	UESSEX
George Zervas	UESSEX
Reza Nejabati	UESSEX
Eduard Escalona	UESSEX

Table of Contents

0 Executive Summary 9

1 Objectives and Scope 10

2 Terminology 12
2.1 Abbreviations 15

3 G^{2}MPLS Routing Extensions 16
3.1 Grid resource modelling in G^{2}MPLS 16
3.1.1 Grid Site description 19
3.1.2 Grid Service description 20
3.1.3 Grid Cluster description 23
3.1.4 Grid Computing Element description 24
3.1.5 Grid Sub-Cluster description 29
3.1.6 Grid Software description 29
3.1.7 Grid Host description 30
3.1.8 Grid Storage Element description 33
3.1.9 Grid Storage Area description 35
3.2 Resource availability calendars 36
3.3 Full-optical TE routing extensions in G^{2}MPLS 39
3.4 G^{2}MPLS opaque extensions for G.I-NNI OSPF (G^2.OSPF-TE) 40
3.4.1 Grid LSA payload details 42
3.4.2 G^{2}MPLS extensions to TE-Link top-level TLV 53
3.5 G^{2}MPLS opaque extensions for G.E-NNI OSPF (G^2.ENNI-OSPF) 55

4 G^{2}MPLS Signalling Extensions 57
4.1 Grid job modelling in G^{2}MPLS 58
4.1.1 Job Definition 60
4.1.2 Job Description 61
4.1.3 Job Identity 62
4.1.4 Job Application 63
4.1.5 Job Resources 64

4.1.6 Job DataStaging description 70
4.2 Special signalling procedures 72
4.2.1 GNS transaction setup 72
4.2.2 Implicit Job destination (anycast) 73
4.2.3 Advance reservations 74
4.2.4 Full-optical TE signalling extensions in G^{2}MPLS 76
4.3 G^{2}MPLS extensions for RSVP-TE (G^2.RSVP-TE) 77
4.3.1 GNS_CALL_EXT Class (C-num = 248) 78
4.3.2 GNS_UNI Class (C-num = 249) 79
4.3.3 Modified Message formats 86

5 Closing notes 91

6 References 92
6.1 Normative references 92
6.2 Informational references 95

7 Acronyms 96

Appendix A Summary of standard GMPLS and OIF TE routing extensions 100

Appendix B Summary of RSVP protocol, base objects and GMPLS extensions 111
B.1 RSVP Common Header 111
B.2 RSVP Message formats 112
B.2.1 Path 112
B.2.2 Resv 113
B.2.3 PathTear 114
B.2.4 ResvTear 114
B.2.5 PathErr 115
B.2.6 ResvErr 115
B.2.7 ResvConf 115
B.2.8 Notify 115
B.2.9 Hello 116
B.2.10 ACK/NACK 116
B.2.11 Srefresh 116
B.3 RSVP Objects 117

B.3.1 SESSION Class (C-num = 1) 117
B.3.2 SESSION ATTRIBUTE Class (C-num = 207) 119
B.3.3 RSVP_HOP Class (C-num = 3) 121
B.3.4 INTEGRITY Class (C-num = 4) 123
B.3.5 TIME_VALUES Class (C-num = 5) 124
B.3.6 ERROR_SPEC Class (C-num = 6) 124
B.3.7 SCOPE Class (C-num = 7) 126
B.3.8 STYLE Class (C-num = 8) 127
B.3.9 FLOWSPEC Class (C-num = 9) 128
B.3.10 FILTER_SPEC Class (C-num = 10) 129
B.3.11 SENDER_TEMPLATE Class (C-num = 11) 131
B.3.12 SENDER_TSPEC Class (C-num = 12) 132
B.3.13 ADSPEC Class (C-num = 13) 133
B.3.14 POLICY_DATA Class (C-num = 14) 134
B.3.15 LABEL Class (C-num = 16) 135
B.3.16 LABEL REQUEST Class (C-num = 19) 136
B.3.17 EXPLICIT ROUTE Class (C-num = 20) 138
B.3.18 RECORD ROUTE Class (C-num = 21) 141
B.3.19 HELLO Class (C-num = 22) 144
B.3.20 MESSAGE_ID Class (C-num = 23) 145
B.3.21 MESSAGE_ID_ACK Class (C-num = 24) 145
B.3.22 MESSAGE_ID_LIST Class (C-num = 25) 146
B.3.23 RECOVERY LABEL Class (C-num = 34) 149
B.3.24 LABEL SET Class (C-num = 36) 149
B.3.25 PROTECTION Class (C-num = 37) 150
B.3.26 SUGGESTED LABEL class (C-num = 129) 151
B.3.27 ACCEPTABLE LABEL SET class (C-num = 130) 151
B.3.28 LSP_TUNNEL_INTERFACE_ID Class (C-num = 193) 151
B.3.29 NOTIFY REQUEST Class (C-num = 195) 152
B.3.30 ADMIN_STATUS Class (C-num = 196) 153
B.3.31 RESTART_CAP Class (C-num = 131) 153
B.3.32 CALL_OPS Class (C-num = 228) 154
B.3.33 GENERALIZED_UNI Class (C-num = 229) 154
B.3.34 CALL_ID Class (C-num = 230) 157
B.4 RSVP Error codes 159

List of Figures

Figure 3-1: Hierarchical object relationship in GLUE schema. 17
Figure 3-2: G^{2}MPLS selection of elements in hierarchical GLUE schema. 18
Figure 3-3: Resource availability schedule (in white). 37
Figure 3-4: Resource availability calendar representation. 37
Figure 3-5: OSPF Opaque LSA. 40
Figure 3-6: OSPF Grid Opaque LSA (type 248). 41
Figure 3-7: Top-level TLV format in OSPF Opaque LSA. 41
Figure 3-8: G.E-NNI 2-layers routing hierarchy. 55
Figure 4-1: JSDL1.0 elements selection for G^{2}MPLS. 60
Figure 4-2: G^{2}RSVP-TE message mapping across the network reference points. 73
Figure 4-3: G^{2}RSVP-TE message sequence for advance reservation signalling and activation. 76

List of Tables

Table 3-1: GLUE Site element and G^{2}MPLS selection of properties. 20
Table 3-2: GLUE Service element and G^{2}MPLS selection of properties. 21
Table 3-3: GLUE Cluster element and G^{2}MPLS selection of properties. 24
Table 3-4: GLUE Computing Element and G^{2}MPLS selection of properties. 26
Table 3-5: GLUE Sub-Cluster element and G^{2}MPLS selection of properties. 29
Table 3-6: GLUE Software element and G^{2}MPLS selection of properties. 30
Table 3-7: GLUE Host element and G^{2}MPLS selection of properties. 31
Table 3-8: GLUE Storage Element and G^{2}MPLS selection of properties. 33
Table 3-9: GLUE Storage Area and G^{2}MPLS selection of properties. 35
Table 3-10: IANA policies for OSPF Opaque Type assignment 40
Table 3-11: TE LSA top level TLVs. 42
Table 3-12: Grid LSA top level TLVs. 42
Table 3-13: Grid Site sub-TLVs. 43
Table 3-14: Grid Service sub-TLVs. 44
Table 3-15: Grid Computing Element sub-TLVs. 46
Table 3-16: Grid Sub-Cluster sub-TLVs. 49
Table 3-17: Grid Storage Element sub-TLVs. 51
Table 3-18: TE-link additional sub-TLVs. 53
Table 4-1: JSDL Job Definition and G^{2}MPLS selection of attributes. 60
Table 4-2: JSDL Job Description and G^{2}MPLS selection of attributes 61
Table 4-3: JSDL Job Identity and G^{2}MPLS selection of attributes. 62
Table 4-4: JSDL Application and G^{2}MPLS selection of attributes. 63
Table 4-5: JSDL Resource and G^{2}MPLS selection of attributes. 67
Table 4-6: JSDL FileSystem and G^{2}MPLS selection of attributes 68
Table 4-7: JSDL OperatingSystem and G^{2}MPLS selection of attributes 69
Table 4-8: JSDL Data Staging and G^{2}MPLS selection of attributes 71
Table 4-9: IANA policy for C-Num allocation in RVSP. 78

0 Executive Summary

This document describes the Grid extensions for the GMPLS routing and signaling protocols.

In section 1 the objectives of the G^{2}MPLS signalling and routing extensions are identified, as well as their positioning with respect to standard protocol objects and procedures. GMPLS and ASON routing and signalling specifications, OIF UNI and OIF E-NNI represent the base reference for these extensions.

In section 2 the relevant terminology is presented by specifying the main sources of this information.

Section 3 describes the Grid extensions for GMPLS routing protocols, by focusing on the modelling of Grid resources in terms of capabilities and availability. Main reference for this work is the GLUE v1.3 schema. Specific protocol objects are presented for G.OSPF-TE, which is the most common IGP used in GMPLS networks and the IGP developed in the G^{2}MPLS prototype. Any IS-IS extension and use in G^{2}MPLS is left for further study.

Section 4 details the extensions to GMPLS signalling protocols, with main reference to the Grid Job Submission Description Language v1.0 (JSDL, GFD-056 revision 14) for the modelling of the job setup phase. Specific protocol objects are presented for G.RSVP-TE, which is the most common protocol for explicit route signalling in GMPLS networks and, therefore, the one developed in the G^{2}MPLS prototype. Any G.CR-LDP extension and use in G^{2}MPLS is left for further study.

In section 5 some concluding remarks are provided.

Summaries of standard G.OSPF-TE and G.RSVP-TE protocol objects, messages and error codes are provided in the document appendices for reference.

1 Objectives and Scope

This document specifies the extensions for routing and signalling protocols in the Grid-enabled GMPLS architecture (a.k.a. G^{2}MPLS).

G^{2}MPLS is a Network Control Plane (NCP) architecture that implements the concept of Grid Network Services (GNS). In the PHOSPHORUS framework, GNS is a service that allows the provisioning of network and Grid resources in a single-step, though a set of seamlessly integrated procedures. As per OFG Open Grid Service Architecture (OGSA, [OGF-GFD80]), GNS belongs to the class of base resources, i.e. those physical or logical resources that are out of the context of the OGSA. Examples of such entities include CPUs and memory in the physical case and licenses, contents and OS processes in the logical case.

G^{2}MPLS architecture is expected to expose interfaces specific for Grids and is made of a set of extensions to the standard ASON/GMPLS architecture. Extensions described in this document have a different use and scope that depends on the network interface (G.OUNI, G.I-NNI, and G.E-NNI) and the Phosphorus model deployed (Overlay or Integrated).

In Phosphorus Overlay, the service discovery and availability extensions for Grid sites are just flooded by routing and piggybacked by signalling at the different interfaces, because of the main scope of G^{2}MPLS Network Control Plane is narrowed to Network Services in this deployment model. In Phosphorus Overlay, Grid information is opaque for G^{2}MPLS routing and signalling, while advance reservations are transparent just for the network resources part.

In Phosphorus Integrated model, Grid information (site capabilities and availabilities) are integral part of routing and signalling decision process, because the co-allocation service are directly implemented by the G^{2}MPLS NCP in this case. Therefore, 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.

Detailed discussion on the architectural background, impact and scope of the extensions here identified can be found in deliverables:

- D2.1 “The Grid-GMPLS Control Plane architecture”,

- D2.6 “Deployment models and solutions of the Grid-GMPLS Control Plane”
- D2.7 ”Grid-GMPLS network interfaces specification - preliminary draft”.

Moreover, this document will not redefine message and objects already part of the current GMPLS and OIF protocols. They are assumed as base reference and mandatory feature list in case of Network Services, which practically make G^{2}MPLS fall back to the standard GMPLS.

2 Terminology

In this section some definitions particularly relevant in the scope of G^{2}MPLS architecture are provided in case with reference to the originator document.

Keyword	Source	Definition
Data model	**[IETF-RFC3444]**	A mapping of the contents of an information model into a form that is specific to a particular type of repository, protocol, platform, etc. It is a rendering of an information model according to a specific set of mechanisms for representing, organizing, storing and handling data. There are typically three parts: • A collection of data structures such as lists, tables, and relations; • A collection of operations that can be applied to the structures such as retrieval, update, and summation; • A collection of integrity rules that define the legal values or changes of state (operations on values). The audience for a data model is implementers.
GLUE schema	**[GLUE]**	An information model that provides a description of core Grid resources at the conceptual level by abstracting real world resources into constructs that can be represented in computer systems (e.g., objects, properties, behaviour, and relationships). The GLUE schema is not tied to any particular implementation and can be profitably used to exchange information among different knowledge domains.
Grid	**[OGF-GFD81]**	A system that is concerned with the integration, virtualization, and management of services and resources in a distributed, heterogeneous environment that supports collections of users and resources (virtual organizations) across traditional administrative and organizational domains (real organizations).
Grid fabric	**[OGF-GFD81]**	The core set of service interfaces that must be implemented in order to realize an OGSA Grid. Also known as the OGSA infrastructure services.

Keyword	Source	Definition
Grid middleware (MW)	**OGF**	Grid technology (a.k.a. middleware) is employed to facilitate formalizing and complying with the Grid context associated with an application execution. Middleware is computer software that connects software components or applications. It is used most often to support complex, distributed applications. It includes web servers, application servers, content management systems, and similar tools that support application development and delivery. Middleware is especially integral to modern information technology based on XML, SOAP, Web services, and service-oriented architecture.
Grid Network Service (GNS)	**[OGF-GNS]**	A network service (e.g. management of QoS classes, policy enforcement points, topology data, network usage metrics, AAA, etc.) with roles and/or interfaces that are deemed to be specific to a Grid infrastructure is a Grid Network Service [OGF-GNS]. Network Services belong to the class of the base resources of the OGSA Architecture [OGSA]. Base resources are those physical or logical resources that are supported entities out of the context of the OGSA. Examples of such entities include CPUs and memory in the physical case and licenses, contents and OS processes in the logical case.
Grid Resource	**[OGF-GFD81]**	In OGSA, a resource is an entity that is useful in a Grid environment. The term usually encompasses entities that are pooled (e.g. hosts, software licenses, IP addresses) or that provide a given capacity (e.g. disks, networks, memory, databases). However, entities such as processes, print jobs, database query results and virtual organizations may also be represented and handled as resources.
Grid service	**[OGF-GFD81]**	In general use, a Grid service is a Web service that is designed to operate in a Grid environment, and meets the requirements of the Grid(s) in which it participates.
Information model	**[IETF-RFC3444]**	An abstraction and representation of entities in a managed environment including properties, operations, and relationships. An information model is independent of implementation: that is, it is protocol-neutral, repository-independent, and platform-independent. An information model's level of specificity is varied, dependent on need. It can be described in a formal language such as UML or an informal natural language such as English. An information model is useful for designers to describe the managed environment, for administrators to understand the modelled objects, and for implementers as a guide to the functionality that can be described, limited by, and coded in the data models. CIM is an example of an object-oriented information model.
Job	**[OGF-GFD81]**	A user-defined task that is scheduled to be carried out by an execution subsystem. In OGSA-EMS, a job is modelled as a manageable resource, has an endpoint reference, and is managed by a job manager.

Keyword	Source	Definition
Job manager	**[OGF-GFD81]**	In OGSA-EMS, a service that manages a set of one or more job instances, which may be structured (e.g. a workflow or dependence graph) or unstructured (e.g. an array of non-interacting jobs). The job manager encapsulates all aspects of job execution, including interacting with execution planning services, the provisioning system, containers, and monitoring services. It may also deal with failures and restarts, it may schedule jobs to resources, and it may collect agreements, reservations and job service data.
Job Submission Description Language (JSDL)	**[OGF-GFD81], [JSDL]**	A language for describing job submissions, including details of their required execution environments. See https://forge.gridforum.org/projects/jsdl-wg for more information.
Nearline storage	**storage architectures**	Nearline is a contraction of Near-online. It is a term used in computer science to describe an intermediate type of data storage. It is a compromise between online storage (constant, very rapid access to data) and offline storage (infrequent access for backup purposes or long-term storage). It is called so because the storage system knows on which volume (cartridge) the data is, and usually asks a robot to retrieve it from his physical location (usually: a tape library) and put it into a tape drive to access it and thus bring the data it contains online. This process is not instantaneous, but it only does require a few seconds, hence the initial description.
Network Control Plane (NCP)	**[ASON-ARCH, ASON-DEF].**	The network Control Plane performs the call control and connection control functions. Through signalling, the Control Plane sets up and releases connections, and may restore a connection in case of a failure. The Control Plane also performs other functions in support of call and connection control, such as routing information dissemination.
TE-link	**[IETF-RFC4201, IETF-RFC4202]**	A traffic engineering (TE) link is a logical construct that represents a way to group/map information about certain physical resources (and their properties) that interconnect LSRs with information that is used by Constrained SPF (for the purpose of path computation) and by GMPLS signalling.
Virtual organization (VO)	**[OGF-GFD81]**	A virtual organization (VO) comprises a set of individuals and/or institutions having direct access to computers, software, data, and other resources for collaborative problem-solving or other purposes. VOs are a concept that supplies a context for operation of the Grid that can be used to associate users, their requests, and a set of resources. The sharing of resources in a VO is necessarily highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs.
Vsite	**UNICORE architecture**	A Vsite identifies a particular set of Grid resources at a UNICORE site (Usite) and is controlled by a Network Job Supervisor (NJS). Vsites may consist of a single supercomputer or a cluster. If more than one resource is operated by an organization there can be one Vsite for each resource inside one Usite.

2.1 Abbreviations

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

3 G²MPLS Routing Extensions

This chapter specifies the routing extensions for Grid-enabled GMPLS.

Additional objects and capabilities here identified are intended to be optional additions to those defined for Traffic Engineering in standard GMPLS routing (IETF RFC4202 and IETF RFC4203 for G.OSPF-TE), ASON routing (ITU-T G.7715 and G.7715.1), and OIF E-NNI routing (OIF ENNI-OSPF-01.0). For this reason, just Grid extensions and Phosphorus enhancements for resource calendars and full-optical TE-links are presented in this chapter. Standard TE extensions for GMPLS and OIF ENNI routing are not presented, though they are fundamental and integral part of any G²MPLS routing implementation. Formatting and composition rules for the standard routing (opaque) objects can be found in the related standards.

As stated in the architecture definition document [PH-WP2-D2.1], G²MPLS adopts a hierarchical routing approach. Actually, this approach is deployed in case of interconnection of neighbouring domains through G²MPLS NCP. On the contrary, the operation of a single G²MPLS 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 E-NNI routing part. G²MPLS 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.

3.1 Grid resource modelling in G²MPLS

Operation of Grids is possible through a key component, the Grid Information Service (GIS), which is responsible for the discovery of capabilities and availabilities of base resources. A common approach in OGSA middleware development is to provide a description of resources as much abstract and implementation-independent as possible. This is the rationale behind GLUE Schema (Grid Laboratory Uniform Environment),

which provides an abstract modelling for Grid resources. The latest stable version of the GLUE Schema is the v1.3, while further extensions towards GLUE Schema v2.0 are under discussion in a specific OGF WG (https://forge.gridforum.org/sf/projects/glue-wg).

In this document, GLUE schema v1.3 is assumed as the main stable reference for Grid resource modelling. GLUE has been preferred among other modelling schemas because of its hierarchical structure that models sites, their supported services, Computing Elements and the Storage Elements (ref. Figure 3-1).

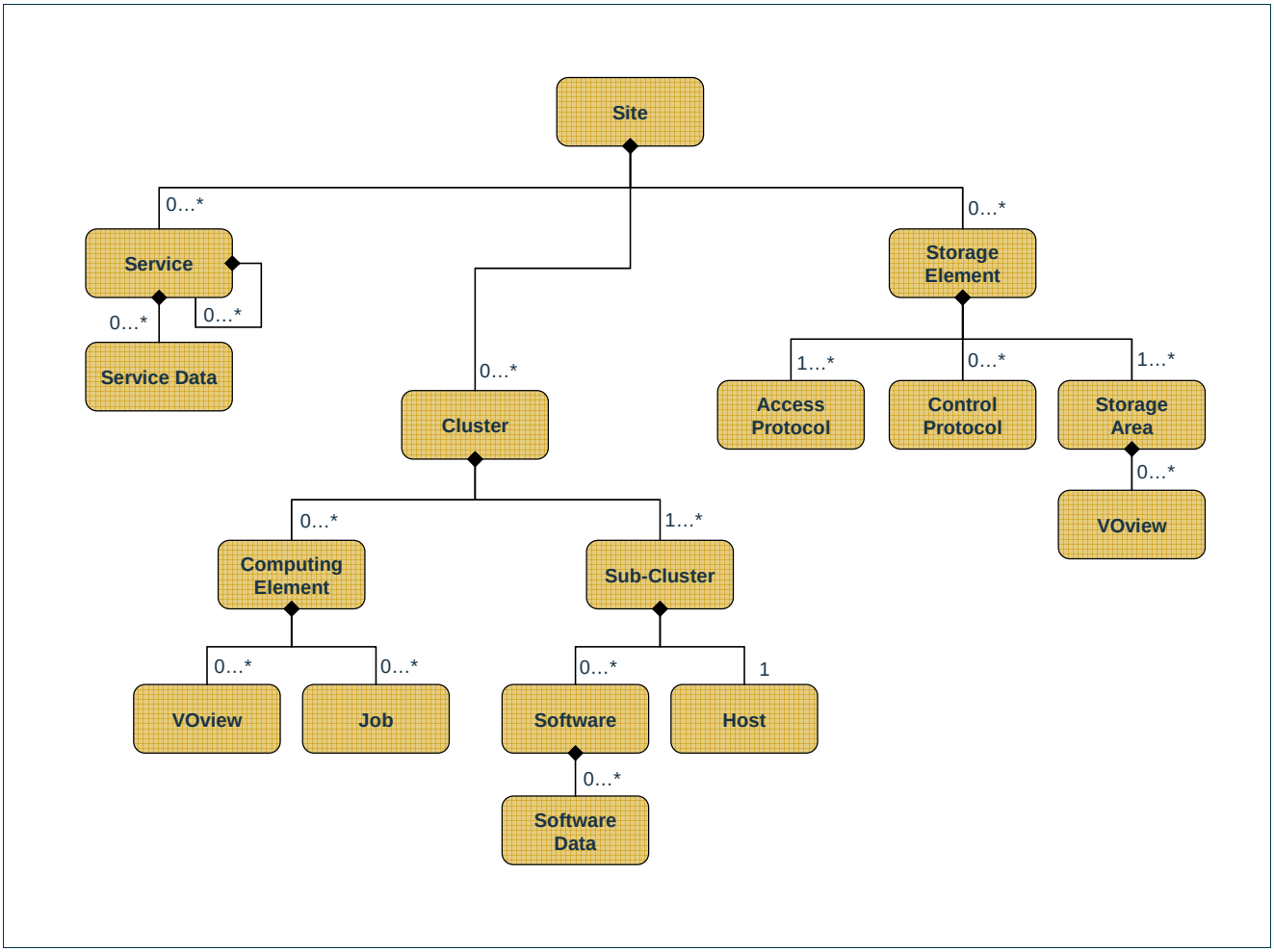


Figure 3-1: Hierarchical object relationship in GLUE schema.

Each GLUE schema object can be detailed with many properties as specified in [GLUE]. Most of these properties are useless for a Control Plane automatic operation and are mainly related to the operation and maintenance of a Grid infrastructure (GIS, Grid Information Service). In both its deployment models, i.e. Phosphorus Overlay and Integrated, G^{2}MPLS is not intended to substitute the GIS, but, on the contrary, G^{2}MPLS will support the Grid middleware towards the creation of single-step reservations for network and Grid resources. Therefore, just a relevant and minimum set of properties has been selected from the GLUE schema,

in order to provide a sufficient detail of information for the Grid elements. The reduced information schema resulting by this selection process (ref. Figure 3-2) maintains the GLUE hierarchy among the elements and tends to achieve the two following objectives:

- to provide the information needed to make the Grid site description consistent
- to limit the size of routing databases and the amount of routing control traffic.

The extra traffic generated by Grid information in GMPLS routing protocol is expected not to hamper G^{2}MPLS node operations, above all if compared with the common low meshing degree of the experimental optical networks used for interconnecting Grid sites. Supporting studies and simulations in WP5 are expected to assess the impact on routing traffic of the proposed extensions in large scale networks. These studies will complement the experimental assessment deriving from the G^{2}MPLS Control Plane tests that will be performed in some local test-beds of the Phosphorus infrastructure.

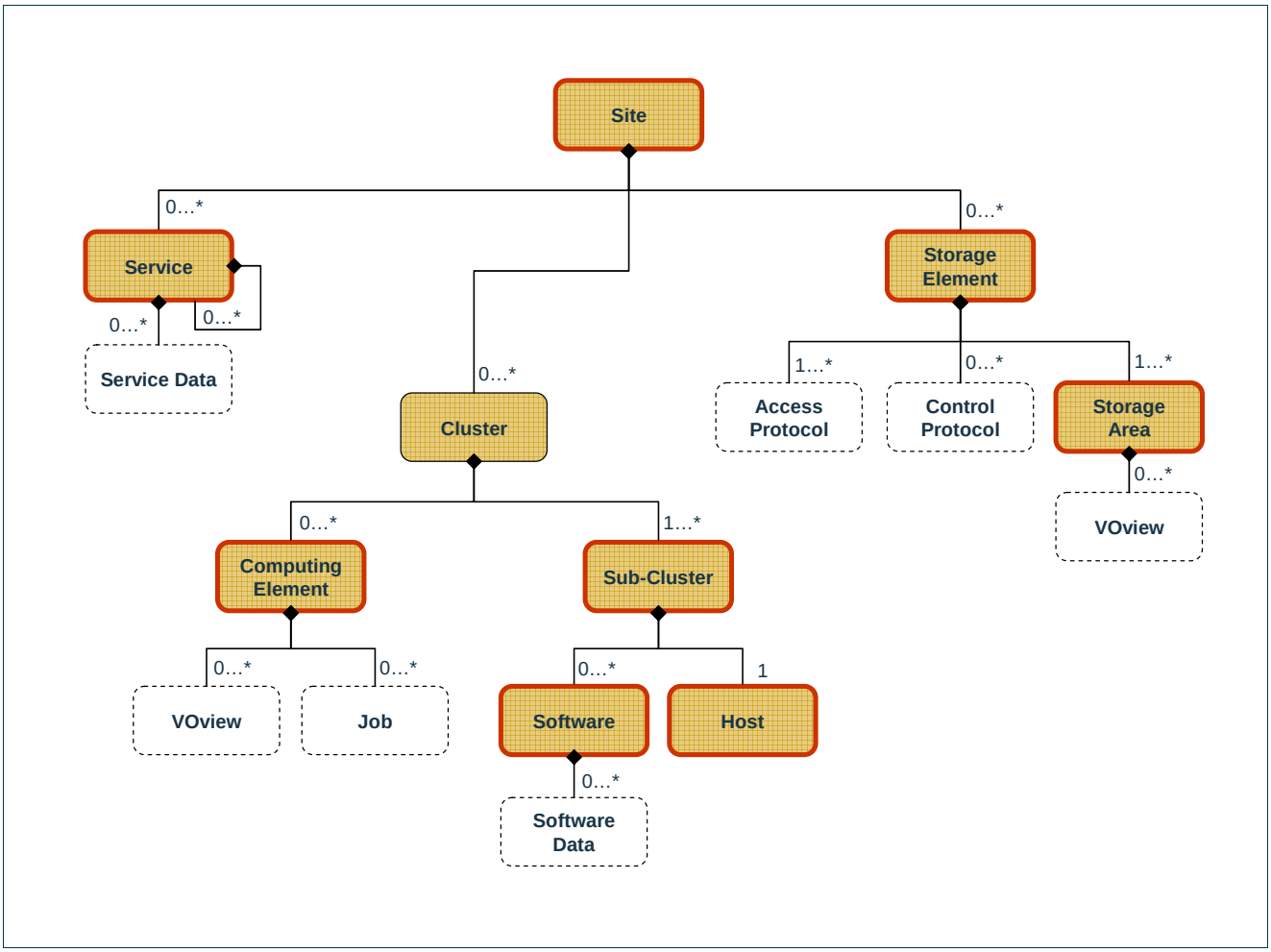


Figure 3-2: G^{2}MPLS selection of elements in hierarchical GLUE schema.

A detailed enumeration of the properties selected from GLUE elements (ref. [GLUE]) for G^{2}MPLS routing is provided in the following sub-sections, as well as the identification of the relevant types and bit fields for G^{2}MPLS routing extensions. The GLUE properties selected for G^{2}MPLS purposes are highlighted in green.

3.1.1 Grid Site description

A Grid Site is a set of resources that are installed and managed by the same organization / set of persons. Selected properties are described in the following Table 3-1.

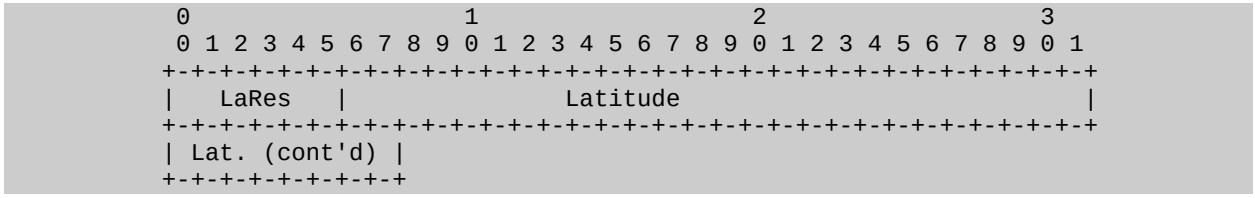
Grid Site Property	Description	Count	GLUE data type	G^{2}MPLS data type
UniqueID	Unique Identifier of the Site	1	string	32 bits
Name	Human-readable name	1	string	string
Description	Short description of this site	1	string	
EmailContact	The main email contact for the site. Syntax rule: "mailto:" followed by a list of email addresses separated by a comma (e.g.: mailto: email1, email2, email3)	1	string	
UserSupportContact	E-mail addresses of the support service. Syntax rule: "mailto:" followed by a list of email addresses separated by a comma (e.g.: mailto: email1, email2, email3)	1	string	
SysAdminContact	E-mail addresses of the support service. Syntax rule: "mailto:" followed by a list of email addresses separated by a comma (e.g.: mailto: email1, email2, email3)	1	string	
SecurityContact	E-mail addresses of the security manager. Syntax rule: "mailto:" followed by a list of email addresses separated by a comma (e.g.: mailto: email1, email2, email3)	1	string	
Location	Geographical location of this site (e.g., city, state, country)	1	string	
Latitude	Degree the position of a place north or south of the equator measured from -90° to +90° with positive values going north and negative values going south	1	real32	40 bits
Longitude	Degree the position of a place east or west of Greenwich, England measured from -180° to +180° with positive values going east and negative values going west	1	real32	40 bits
Web	The URI identifying a web page with more information about this site	1	uri	
Sponsor	VO sponsoring the site; the syntax should allow the expression of the percentage of sponsorship	N	string	

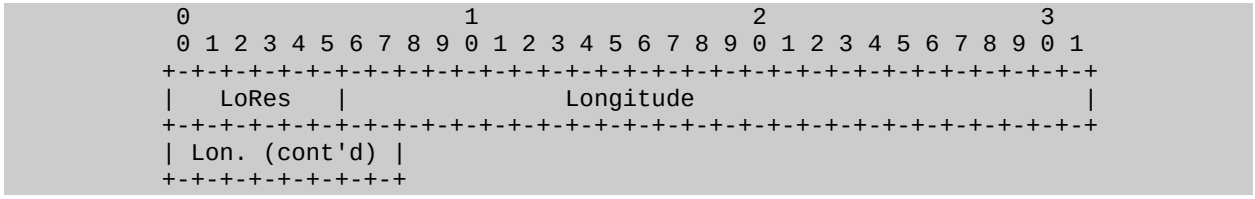
Grid Site Property	Description	Count	GLUE data type	G²MPLS data type
OtherInfo	This attribute is to be used to publish info that does not fit in any other attribute of the site entity. A name=value pair syntax or an XML structure are example of usage.	N	string	

Table 3-1: GLUE Site element and G²MPLS selection of properties.

Grid Site Unique ID (32 bits) is a Grid-wide identifier for the site assigned during the address planning of the infrastructure.

Latitude and Longitude are geographical coordinates. As per IETF RFC3825, they can be represented in fixed-point 2s-complement binary degrees, for the economy of a smaller option size compared to a string encoding of digits. The integer parts of these fields are 9 bits long to accommodate +/- 180 degrees. The fractional part is 25 bits long, better than the precision of 7 decimal digits. The length of each field is 40 bits, 34 of which is the sum of the integer (9) and fractional (25) bits, plus 6 bits of resolution.





3.1.2 Grid Service description

A Grid Service is an abstracted, logical view of actual software components that should be formally defined in terms of the messages exchanged between provider entity and requester entity. Selected properties are described in the following Table 3-2.

Grid Service Property	Description	Count	GLUE data type	G²MPLS data type
UniqueID	Unique Identifier of the Service	1	string	32 bits
Name	Human-readable name	1	string	

Grid Service Property	Description	Count	GLUE data type	G²MPLS data type
Type	The service type.	1	serviceType_t	grid_service_type_t (16 bits)
Version	Version of the service: <major version number>.<minor version number>.<patch version number>	1	string	version_t (16 bits)
Endpoint	Network endpoint for this service	1	uri	IPv4, IPv6, NSAP
Status	Status of the service. Enumeration: OK, Warning, Critical, Unknown, Other.	1	serviceStatus_t	grid_service_status_t (8 bits)
StatusInfo	Textual explanation for the status of the service	1	string	
WSDL	URI of the WSDL describing the service	1	uri	
Semantics	URL of detailed description	1	uri	
StartTime	The timestamp related to last start time of this service	1	dateTime_xs-t	
Owner	Owner of the service (e.g.: one or more VO's)	N	string	
AccessControlBase.Rule	Authorization rule for this entity	N	ACL_t	

Table 3-2: GLUE Service element and G²MPLS selection of properties.

G²MPLS Service Unique ID (32 bits) is an identifier local to the Grid site assigned for the service during the configuration phase of the site.

Endpoint is the address of the endpoint hosting this service and it is in the form of an IPv4 (32 bits), IPv6 (128 bits) or NSAP (160 bits) address. It substitutes the TNA opaque sub-TLV in OSPF-TE E-NNI.

Service Type can assume one of the following values in the grid_service_type_t (16 bits)

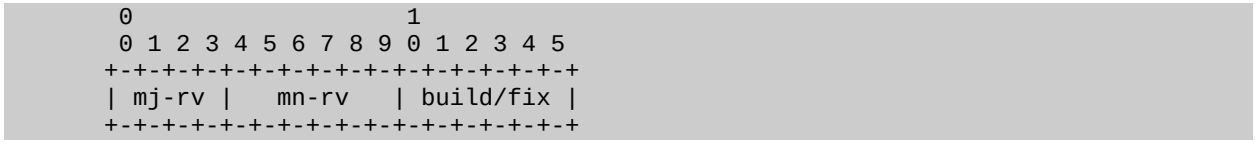
- org.glite.wms 0x0001
- org.glite.rgma.LatestProducer 0x0002
- org.glite.rgma.StreamProducer 0x0003
- org.glite.rgma.DBProducer 0x0004
- org.glite.rgma.CanonicalProducer 0x0005
- org.glite.rgma.Archiver 0x0006
- org.glite.rgma.Consumer 0x0007
- org.glite.rgma.Registry 0x0008
- org.glite.rgma.Schema 0x0009
- org.glite.rgma.Browser 0x000A
- org.glite.rgma.PrimaryProducer 0x000B
- org.glite.rgma.SecondaryProducer 0x000C
- org.glite.rgma.OnDemandProducer 0x000D
- org.glite.voms 0x000E
- org.glite.FiremanCatalog 0x000F

- org.glite.SEIndex 0x0010
- org.glite.Metadata 0x0011
- org.glite.ChannelManagement 0x0012
- org.glite.FileTransfer 0x0013
- org.glite.FileTransferStats 0x0014
- org.glite.ChannelAgent 0x0015
- org.glite.KeyStore 0x0016
- org.glite.FAS 0x0017
- org.glite.gliteIO 0x0018
- SRM 0x0100
- gsiftp 0x0200
- org.edg.local-replica-catalog 0x0300
- org.edg.replica-metadata-catalog 0x0301
- org.edg.SE 0x0302
- it.infn.GridICE 0x0400
- MyProxy 0x0500
- GUMS 0x0600
- GridCat 0x0700
- edu.caltech.cacr.monalisa 0x0800
- OpenSSH 0x0900
- MDS-GIIS 0x0A00
- BDII 0x0B00
- RLS 0x0C00
- data-location-interface 0x0D00
- pbs.torque.server 0x0E00
- pbs.torque.maui 0x0E01
- unicore.core.targetSystemFactory 0x0F01
- unicore.core.targetSystem 0x0F02
- unicore.core.storageManagement 0x0F03
- unicore.core.fileTransfer 0x0F04
- unicore.core.jobManagement 0x0F05
- unicore.core.registry 0x0F06
- unicore.workflow.workflowFactory 0x0F07
- unicore.workflow.workflowManagement 0x0F08
- unicore.workflow.serviceOrchestrator 0x0F09
- unicore.workflow.gridResourceInformationService 0x0F0A
- unicore.CISInformationProvider 0x0F0B
- other 0XFF00

Listed services derive mainly from the GLUE schema and are related to operational Grid infrastructures using GLUE for their GIS. GLOBUS core services are included originally in the service list. However, services highlighted in yellow box are related to UNICORE 6, which does not support GLUE schema v1.3. These last

services must be considered Phosphorus specific additions and will be the most common services exported by sites in the test-bed assessment phase jointly with the GLOBUS core services.

Version (16 bits) provides the specific version of the service in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits).



Status can assume one of the following values in the grid_service_status_t (4 bits)

- Unknown 0x0
- OK 0x1
- Warning 0x2
- Critical 0x3
- Other 0xF

3.1.3 Grid Cluster description

A Grid Cluster is a set of machines providing computing power managed by a local management system. Selected properties are described in the following Table 3-3.

Grid Cluster Property	Description	Count	GLUE data type	G²MPLS data type
UniqueID	Unique ID associated to the cluster (typically refers to the host name of the machine where the LRMS runs)	1	string	
Name	Name of the cluster. It does not need to be unique and can be used as a human-readable name	1	string	
TmpDir	The path of a temporary directory shared across worker nodes. This directory must be available to programs using the normal file access primitives (open/read/write/close) and possibly provide some lock mechanisms. The view provided by this directory on different hosts must be synchronized (e.g., if a host writes some content in a file, a read operation from a different host must be able to access that content. It may be necessary to call an explicit synchronization primitive, depending on the technology used). This directory may be used as shared space by programs running on multiple hosts (e.g. MPI)	1	string	

Grid Cluster Property	Description	Count	GLUE data type	G²MPLS data type
WNTmpDir	The path of a temporary directory local to each Worker Node. This directory must be POSIX compliant. This will probably be the run directory for jobs running on that WN. Applications must be able to perform all the operations supported on local disks by that OS (e.g. open/read/write files or special files like pipes, create locks and change permissions). The Cluster or the CE may take care of providing an empty directory for the job and remove the directory once the job finished.	1	string	

Table 3-3: GLUE Cluster element and G²MPLS selection of properties.

The level of abstraction provided by the Cluster container is too coarse for any Grid resource modelling to be used in co-allocation services. In fact, the possible bundling of information from its component elements, i.e. Computing Elements and Sub-Clusters, could result in a limit for the flexible and scalable advertisement of attribute changes. Therefore, in G²MPLS the Grid Cluster is described through its component.

3.1.4 Grid Computing Element description

A Grid Computing Element is a service that manages jobs and offers them execution environments provided by computing resources. The considered computing resources are those assigned to a single batch queue. Selected properties are described in the following Table 3-4.

Grid Computing Element Property	Description	Count	GLUE data type	G²MPLS data type
UniqueID	Unique Identifier for this computing element	1	string	32 bits
Name	The name of the underlying batch queue	1	string	
ImplementationName	The name of the implementation	1	CEImpl_t	
ImplementationVersion	The version of the implementation	1	string	
Capability	General way to advertise functions supported by this entity	n	string	
InformationServiceURL	Contact URI of the service providing for status and characteristics information (e.g., the URI of the MDS GRIS that is the primary source for the class instance information, this is useful to locate such an endpoint from a top-level GIIS)	1	uri	
Info.LRMSType	Type of the underlying local resource management system	1	lrms_t	grid_lrms_t (16 bits)
Info.LRMSVersion	Version of the local resource management system	1	string	version_t (16 bits)

Grid Computing Element Property	Description	Count	GLUE data type	G²MPLS data type
Info.HostName	Host name of the machine running this service	1	string	IPv4, IPv6, NSAP
Info.GatekeeperPort	Gatekeeper port	1	int32	32 bits
Info.JobManager	The job manager used by the gatekeeper (e.g.: jobmanager-pbs). Generally speaking, it is a string that allows to distinguish between different queues accessible using the same host and port.	1	string	string
Info.ContactString	String specifying how to contact the service. A default value can be HostName:GatekeerPort/Jobmanager. It identifies an endpoint for computing resources in a given protocol (usually GRAM)	n	string	
Info.ApplicationDir	The path of the directory available for application installation. Typically a POSIX accessible disk space with transient to permanent allocation to the users	1	string	
Info.DataDir	The path of a shared directory available for application data. Typically a POSIX accessible transient disk space shared between the Worker Nodes. It may be used by MPI applications or to store intermediate files that need further processing by local jobs or as staging area, specially if the Worker Node have no internet connectivity	1	string	string
Info.DefaultSE	Unique identifier of the default Storage Element. Unique identifier of the default Storage Element to be used to store files from jobs in the CE in cases where no destination SE is explicitly stated	1	uri	32 bits
State.Status	The queue status: • `Queueing' the queue can accept job submission, but can not be served by the scheduler; • `Production' the CE can accept job submissions and is served by a scheduler; • `Closed' The CE can not accept job submission and can not be served by a scheduler; • `Draining' the CE can not accept job submission, but can be served by a scheduler.	1	ce_status_t	grid_cese_status_t (8 bits)
State.RunningJobs	The number of jobs in running state	1	int32	32 bits
State.WaitingJobs	The number of jobs in waiting state	1	int32	32 bits
State.TotalJobs	The number of jobs in any state	1	int32	32 bits
State.EstimatedResponseTime	Based on the accepted jobs, estimated time to last for a new job from the acceptance to the start of its execution	1	int32	32 bits
State.WorstResponseTime	Among the accepted jobs, the worst time from the job being accepted by the service to the start of its execution	1	int32	32 bits

Grid Computing Element Property	Description	Count	GLUE data type	G²MPLS data type
State.FreeJobSlots	Number of free job slots (sometimes called logical CPUs), i.e., number of single-processor jobs which could be started if no other jobs are submitted and no jobs finish in the interim. This could be the size of the queue less the used resources, but a policy could influence this number	1	int32	16 bits
Policy.MaxWallClockTime	The default maximum wall clock time allowed to each job by the batch system if no limit is requested. Once this time has expired the job will most likely be killed or removed from the queue	1	int32	32 bits
Policy.MaxObtainableWallClockTime	The maximum obtainable wall clock time that can be granted to the job upon user request	1	int32	32 bits
Policy.MaxCPUTime	The default maximum CPU time allowed to each job by the batch system	1	int32	32 bits
Policy.MaxObtainableCPUTime	The maximum obtainable CPU time that can be granted to the job upon user request	1	int32	32 bits
Policy.MaxTotalJobs	The maximum allowed number of jobs in the CE	1	int32	32 bits
Policy.MaxRunningJobs	The maximum allowed number of jobs in running state in the CE	1	int32	32 bits
Policy.MaxWaitingJobs	The maximum number of jobs that can be in waiting state	1	int32	32 bits
Policy.Priority	The priority given to jobs in this CE. The lower the number, the higher the priority	1	int32	7 bits
Policy.AssignedJobSlots	Number of slots for jobs to be in running state (it represents the maximum number of single-processor jobs that can be running at a given time)	1	int32	16 bits
Policy.MaxSlotsPerJobs	The maximum number of slots which could be allocated to a single job (defined to be 1 for a site accepting only standard jobs).	1	int32	16 bits
Policy.Preemption	If true, the batch system enables preemption of jobs	1	boolean	1 bit
AccessControlBase.Rule	Authorization rule for this entity	1	ACL_t	

Table 3-4: GLUE Computing Element and G²MPLS selection of properties.

The Grid Computing Element ID (32 bits) is an identifier local to the cluster assigned during the configuration phase of the site.

HostName is the address of the machine running this service and it should be in the form of and IPv4, IPv6 or NSAP address.

GatekeeperPort (32 bits) is the port used by the gatekeeper (in any) in the Grid site.

LRMSType can assume one of the following values in the grid_lrms_t (16 bits)

- OpenPBS 0x0001
- LSF 0x0002
- Condor 0x0003
- BQS 0x0004
- CondorG 0x0005
- FBSNG 0x0006
- Torque 0x0007
- PBSPro 0x0008
- SGE 0x0009
- NQE 0x000A
- fork 0x000B
- other 0xFFFF

LRMSVersion (16 bits) is the version of the local resource management system specified in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| mj-rv |   mn-rv   | build/fix |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

JobManager is a string identifying the job manager used by the gatekeeper (e.g.: jobmanager-pbs). Generally speaking, it is a string that distinguishes between different queues accessible using the same host and port.

DataDir is a POSIX compliant string representing the path of a run directory for jobs running on that Working Node.

DefaultSE (32 bits) is the unique identifier of the default Storage Element for this computing element. Unique identifier of the default Storage Element to be used to store files from jobs in the CE in cases where no destination SE is explicitly stated.

Status represents the queue status and can assume one of the following values in the grid_cese_status_t (4 bits)

- `Queuing' (0x01), the queue can accept job submission, but can not be served by the scheduler;
- `Production' (0x02), the CE can accept job submissions and is served by a scheduler;
- `Closed' (0x03), the CE can not accept job submission and can not be served by a scheduler;
- `Draining' (0x04), the CE can not accept job submission, but can be served by a scheduler.

State.RunningJobs (32 bits) is the number of jobs in running state.

State.WaitingJobs (32 bits) is the number of jobs in waiting state

State.TotalJobs (32 bits) is the number of jobs in any state.

State.EstimatedResponseTime (32 bits) is the estimated time to last for a new job from the acceptance to the start of its execution.

State.WorstResponseTime (32 bits) is the worst time from the job being accepted by the service to the start of its execution.

State.FreeJobSlots (16 bits) is the number of free job slots (sometimes called logical CPUs), i.e., number of single-processor jobs which could be started if no other jobs are submitted and no jobs finish in the interim. This could be the size of the queue less the used resources, but a policy could influence this number.

Policy.MaxWallClockTime (32 bits) is the default maximum wall clock time allowed to each job by the batch system if no limit is requested. Once this time has expired the job will most likely be killed or removed from the queue

Policy.MaxObtainableWallClockTime (32 bits) is the maximum obtainable wall clock time that can be granted to the job upon user request.

Policy.MaxCPUTime (32 bits) is the default maximum CPU time allowed to each job by the batch system.

Policy.MaxObtainableCPUTime (32 bits) is the maximum obtainable CPU time that can be granted to the job upon user request.

Policy.MaxTotalJobs (32 bits) is the maximum allowed number of jobs in the Computing Element.

Policy.MaxRunningJobs (32 bits) is the maximum allowed number of jobs in running state in the CE.

Policy.MaxWaitingJobs (32 bits) is the maximum number of jobs that can be in waiting state.

Policy.Priority (7 bits) is a priority value given to jobs in this CE. The lower the number, the higher the priority

Policy.AssignedJobSlots (16 bits) is the number of slots for jobs to be in running state (it represents the maximum number of single-processor jobs that can be running at a given time).

Policy.MaxSlotsPerJobs (16 bits) is the maximum number of slots which could be allocated to a single job (defined to be 1 for a site accepting only standard jobs).

Policy.Preemption (1 bit) is a flag indicating if the batch system enables pre-emption of jobs or not.

> All the time values in the Computing Element are in the form of 32-bit unsigned fixed-point numbers, in seconds relative to 0h on 1 January 1900. These are NTP reduced timestamps (ref. IETF RFC 4330). This field will overflow some time in 2036 (second 4,294,967,296). Similar actions as per NTP could be taken in that case if the proposed extension will be still in use.

3.1.5 Grid Sub-Cluster description

A Grid Sub-Cluster contains information about a homogeneous set of hosts as regards a selected number of host attributes. Selected properties are described in the following Table 3-5.

Grid Sub-Cluster Property	Description	Count	GLUE data type	G²MPLS data type
UniqueID	Unique ID of the sub-cluster	1	string	32 bits
Name	Name of the sub-cluster	1	string	
PhysicalCPUs	The total number of real CPUs in the sub-cluster	1	int32	16 bits
LogicalCPUs	The effective number of CPUs in the sub-cluster, including the effect of hyper-threading and the effects of virtualization due to the queuing system	1	int32	16 bits

Table 3-5: GLUE Sub-Cluster element and G²MPLS selection of properties.

The Grid Sub-Cluster ID (32 bits) is an identifier local to the cluster assigned during the configuration phase of the site.

PhysicalCPUs (16 bits) accounts for the total number of real CPUs in the sub-cluster.

LogicalCPUs (16 bits) is the effective number of CPUs in the sub-cluster, including the effect of hyper-threading and the effects of virtualization due to the queuing system.

3.1.6 Grid Software description

A Grid Software contains information about an installed software package. Selected properties are described in the following Table 3-6.

Grid Software Property	Description	Count	GLUE data type	G²MPLS data type
LocalID	local identifier for the location (suggested value: concatenation of Name and Version attributes separated by the + character)	1	string	grid_software_t (16 bits)
Name	A name for this software package	1	string	
Version	Version, following the syntax adopted by the software	1	string	version_t (16 bits)
InstalledRoot	The directory where the software is installed on the file system	1	string	
EnvironmentSetup	Fully qualified script for the setting of the application environment	1	string	string

Grid Software Property	Description	Count	GLUE data type	G²MPLS data type
ModuleName	The name of the module that gets loaded (to set the environment) before the job runs	1	string	

Table 3-6: GLUE Software element and G²MPLS selection of properties.

LocalID (16 bits) is a local identifier for the software name and can assume one of the following values in the grid_software_t

- Wisdom 0x0001
- Kodavis 0x0002
- TOPS 0x0003
- DDSS 0x0004
- INCA 0x0005
- other 0xFFFF

EnvironmentSetup is a string representing a fully qualified script for the setting of the application environment.

Version (16 bits) provides the specific version of the service in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| mj-rv |   mn-rv   | build/fix |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.7 Grid Host description

A Grid Host is fundamental part of the sub-cluster and information here reported extends the attributes of the sub-cluster. Selected properties are described in the following Table 3-7.

Grid Host Property	Description	Count	GLUE data type	G²MPLS data type
OperatingSystemName	Name of the operating system	1	string	grid_os_t (16 bits)
OperatingSystemRelease	Release of the operating system	1	string	version_t (16 bits)
OperatingSystemVersion	Version of the operating system	1	string	
ProcessorModel	Processor model as defined by the vendor	1	string	grid_cpu_arch_t (8 bits)
ProcessorVersion	Processor version	1	string	
ProcessorVendor	Name of the processor vendor	1	string	

Grid Host Property	Description	Count	GLUE data type	G²MPLS data type
ProcessorClockSpeed	MHz The clock speed	1	int32	
ProcessorInstructionSet	The processor instruction set; use comma-separated values (e.g.: mmx, cisc)	1	string	
ProcessorOtherDescription	Other description for the processor	1	string	
MainMemoryRAMSize	The amount of RAM in MB	1	int32	32 bits
MainMemoryVirtualSize	The amount of Virtual Memory (RAM+Swap) in MB	1	int32	32 bits
NetworkAdapterOutboundIP	Permission for direct outbound connectivity, even if limited	1	boolean	
NetworkAdapterInboundIP	Permission for inbound connectivity, even if limited	1	boolean	
ArchitecturePlatformType	Platform type of the host	1	string	
ArchitectureSMPSize	number of physical CPUs in the host	1	int32	
BenchmarkSI00	SpecInt2000	1	int32	
BenchmarkSF00	SpecFloat2000	1	int32	
App.Soft.RunTimeEnvironment	ApplicationSoftwareRunTimeEnvironment: environment variable associated to an installed software package	n	string	

Table 3-7: GLUE Host element and G²MPLS selection of properties.

The processorInfo (8 bits) can assume one of the following values in the grid_cpu_arch_t:

- SPARC 0X01
- POWERPC 0X02
- X86 0X03
- X86_32 0X04
- X86_64 0X05
- PARISC 0X06
- MIPS 0X07
- IA64 0X08
- ARM 0X09
- other 0XFF

MainMemoryRAMSize (32 bits) is the amount of RAM in MB.

MainMemoryVirtualSize (32 bits) is the amount of Virtual Memory (RAM+Swap) in MB.

The operatingSystemName (16 bits) can assume one of the following values in the grid_os_t

- Unknown 0x0000
- MACOS 0x0001
- ATTUNIX 0x0002
- DGUX 0x0003
- DECNT 0x0004
- Tru64_UNIX 0x0005
- OpenVMS 0x0006
- HPUX 0x0007
- AIX 0x0008
- MVS 0x0009
- OS400 0x000A
- OS_2 0x000B
- JavaVM 0x000C
- MSDOS 0x000D
- WIN3x 0x000E
- WIN95 0x000F
- WIN98 0x0010
- WINNT 0x0011
- WINCE 0x0012
- NCR3000 0x0013
- NetWare 0x0014
- OSF 0x0015
- DC_OS 0x0016
- Reliant_UNIX 0x0017
- SCO_UnixWare 0x0018
- SCO_OpenServer 0x0019
- Sequent 0x001A
- IRIX 0x001B
- Solaris 0x001C
- SunOS 0x001D
- U6000 0x001E
- ASERIES 0x001F
- TandemNSK 0x0020
- TandemNT 0x0021
- BS2000 0x0022
- LINUX 0x0023
- Lynx 0x0024
- XENIX 0x0025
- VM 0x0026
- Interactive_UNIX 0x0027
- BSDUNIX 0x0028
- FreeBSD 0x0029
- NetBSD 0x002A
- GNU_Hurd 0x002B
- OS9 0x002C
- MACH_Kernel 0x002D
- Inferno 0x002E
- QNX 0x002F
- EPOC 0x0030
- IxWorks 0x0031
- VxWorks 0x0032
- MiNT 0x0033
- BeOS 0x0034
- HP_MPE 0x0035
- NextStep 0x0036
- PalmPilot 0x0037
- Rhapsody 0x0038
- Windows_2000 0x0039
- Dedicated 0x003A
- OS_390 0x003B
- VSE 0x003C
- TPF 0x003D
- Windows_R_Me 0x003E
- Caldera_Open_UNIX 0x003F
- OpenBSD 0x0040
- Not_Applicable 0x0041
- Windows_XP 0x0042
- z_OS 0x0043
- other 0xFFFF

The operatingSystemVersion (16 bits) provides the specific version of an OS in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| mj-rv |   mn-rv   | build/fix |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.8 Grid Storage Element description

The Grid Storage Element contains an abstraction for a storage resource, group of services, protocols and data sources. Selected properties are described in the following Table 3-8.

Grid Storage Element Property	Description	Count	GLUE data type	G²MPLS data type
UniqueID	Unique Identifier of the Storage Element	1	string	32 bits
Name	Human-friendly name for the SE	1	string	
Architecture	Underlying architectural system category. String enumeration: disk, tape, multi-disk, other	1	SEArch_t	grid_storage_arch_t (4 bits)
InformationServiceURL	URL of the information service providing details for this SE (e.g., the URI of the MDS GRIS that is the primary source for the class instance information, this is useful to locate such an endpoint from a top-level GIIS)	1	url	
ImplementationName	The name of the implementation	1	SEImple_t	
ImplementationVersion	The version of the implementation	1	string	
Status	The status of the whole SE: `Queuing' the SE can accept new requests, but they will be kept on hold; `Production' the SE processes old and new requests according to its policies; `Closed' the SE does not accept new requests and does not process old requests; `Draining' the SE does not accept new requests, but still processes old requests	1	SEStatus_t	grid_cese_status_t (4 bit)
TotalOnlineSize	Total size of online storage space (in GB)	1	int32	32 bits
TotalNearlineSize	Total size of nearline storage (in GB)	1	int32	32 bits
UsedOnlineSize	Used online storage (in GB)	1	int32	32 bits
UsedNearlineSize	Used nearline storage (in GB)	1	int32	32 bits
Access Protocol	Protocols available to access/transport files in/from storage areas	n	accessProt_t	12 bits bitmask
Control Protocol	Protocol available for the control and/or management of the storage resource	n	ControlProt_t	12 bits bitmask

Table 3-8: GLUE Storage Element and G²MPLS selection of properties.

The Grid SE Unique ID (32 bits) is an identifier local to the Grid site assigned during the configuration phase of the site.

Architecture (4bits) identifies the underlying architectural system category according to the enumerated g2mpls_ceSeAarch_t

- disk 0x1

- tape 0x2
- multidisk 0x3
- other 0xF

Status represents the storage status and can assume one of the following values in the grid_cese_status_t (4 bits)

- “Queueing” (0x1), the storage can accept new requests, but they are kept on hold;
- “Production” (0x2), the SE processes old and new requests according to its policies;
- “Closed” (0x3), the SE does not accept new requests and does not process old requests;
- “Draining” (0x4), the SE does not accept new requests, but still processes old requests.

TotalOnlineSize (32 bits) is the total size of online storage space in GB.

TotalNearlineSize (32 bits) is the total size of nearline storage in GB.

UsedOnlineSize (32 bits) is the used online storage in GB.

UsedNearlineSize (32 bits) is the used nearline storage in GB.

Access Protocol (12 bits) is a bitmask of protocols available to access/transport files in/from storage areas. More than one protocol can be present for an SE.

- bit 01 → gsiftp
- bit 02 → nfs
- bit 03 → afs
- bit 04 → rfio
- bit 05 → gsirfio
- bit 06 → dcap
- bit 06 → gsidcap
- bit 08 → root
- bit 09 → https

Control Protocol (12 bits) is a bit-mask of the protocols available for the control and/or management of the storage resource. More than one protocol can be present for an SE.

- bit 01 → SRM
- bit 02 → org.edg.SE
- bit 03 → classic

3.1.9 Grid Storage Area description

The Grid Storage Area represents a portion of storage extent to which a uniform set of policies applies. Selected properties are described in the following Table 3-9.

Grid Storage Area Property	Description	Count	GLUE data type	G²MPLS data type
Name	Human-friendly name for the area	1	string	string
Path	Full path of the root directory for this storage area	1	string	string
AccessControlBase.Rule	Authorization rule for this entity	1	ACL_t	
TotalOnlineSize	Total online storage space (in GB)	1	int32	32 bits
UsedOnlineSize	Used online storage space (in GB)	1	int32	
FreeOnlineSize	Free online storage space (in GB)	1	int32	32 bits
ReservedOnlineSize	Reserved online storage space (in GB)	1	int32	32 bits
TotalNearlineSize	Total nearline storage space (in GB)	1	int32	32 bits
UsedNearlineSize	Free nearline storage (in GB)	1	int32	
FreeNearlineSize	Free nearline storage space (in GB)	1	int32	32 bits
ReservedNearlineSize	Reserved nearline storage space (in GB)	1	int32	32 bits
RetentionPolicy	Possible values: custodial, output, replica	1	retentionPol_t	grid_retention_policy_t (4 bits)
AccessLatency	values: online, nearline, offline	1	accessLat_t	grid_access_latency_t (4 bits)
ExpirationMode	Possible values: neverExpire, warnWhenExpired, releaseWhenExpired	1	expirationMode_t	grid_expiration_mode_t (4 bits)
Capability	General way to advertise functions supported by this entity	n	string	

Table 3-9: GLUE Storage Area and G²MPLS selection of properties.

Name is a string representing a human-friendly name for the storage area.

TotalOnlineSize (32 bits) is the total online storage space in GB.

FreeOnlineSize (32 bits) is the free online storage space in GB.

ReservedOnlineSize (32 bits) is the reserved online storage space in GB.

Both FreeOnlineSize and ReservedOnlineSize cannot exceed TotalOnlineSize both singularly and added up.

TotalNearlineSize (32 bits) is the total nearline storage space in GB.

FreeNearlineSize (32 bits) is the free nearline storage space in GB.

ReservedNearlineSize (32 bits) is the reserved nearline storage space in GB.

Also FreeNearlineSize and ReservedNearlineSize cannot exceed TotalNearlineSize both singularly and added up.

RetentionPolicy (4 bit) identifies the action on data stored according to the following possible values:

- custodial 0x1,
- output 0x2,
- replica 0x3

AccessLatency (4 bit) identifies the type of latency while accessing data according to the following possible values:

- online 0x1,
- nearline 0x2,
- offline 0x3,

ExpirationMode (4 bit) identifies the type of expiration of the stored data according to the following possible

- neverExpire 0x1,
- warnWhenExpired 0x2,
- releaseWhenExpired 0x3,

3.2 Resource availability calendars

Advance reservation process is based on the availability of information about schedules of resources. In this context a resource could be one element of a Grid site as described in sec. 3.1 or a TE-link (i.e. a pure network Transport Plane resource).

Time availability is described in the form of calendars, whose simplest implementation is shown in Figure 3-3 and Figure 3-4. Availability calendars are ordered list of pairs and each pair contains:

- a time value, expressed in NTP-reduced format[1] (32 bit just for seconds) identifying when resource availability is going to change,
- a new resource availability value.

```
ResourceCalendar = [ (T1, A1), (T2, A2), ..., ..., (Tn, An) ]
```

[1] All the time values are in the form of 32-bit unsigned fixed-point numbers, in seconds relative to 0h on 1 January 1900. These are NTP reduced timestamps (ref. IETF RFC4330). This field will overflow some time in 2036 (second 4,294,967,296). Similar actions as per NTP could be taken in that case if the proposed extension will be still in use.

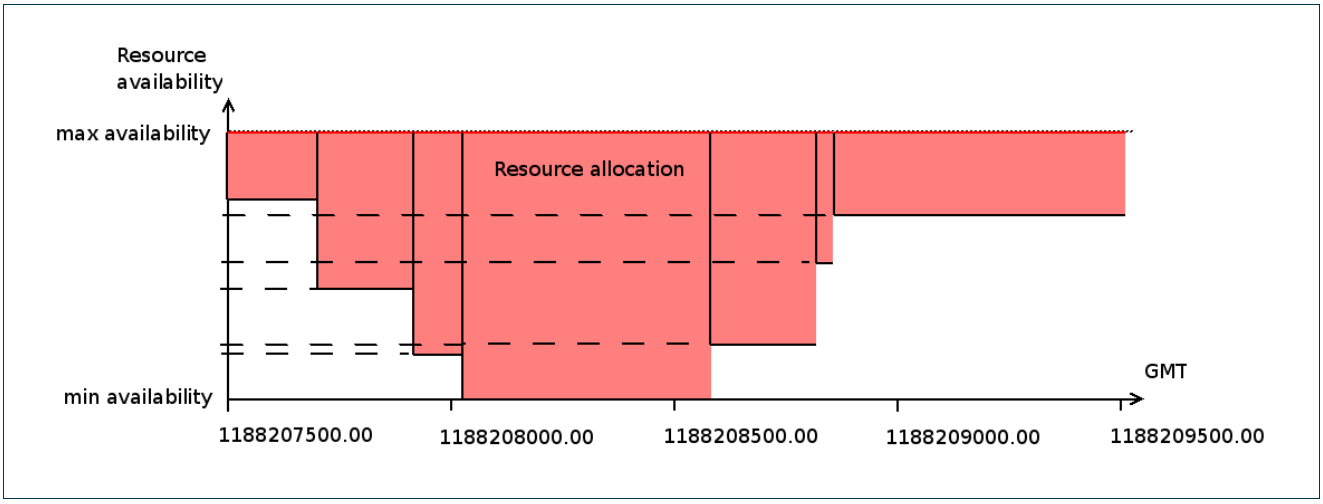


Figure 3-3: Resource availability schedule (in white).

The depth of the availability calendars, i.e. their projection in the future, is limited to a threshold of no more than 10 events in order to limit the size of the object and its replica into each routing database. It is up to the originating node to update the calendar once the time of the older event expires.

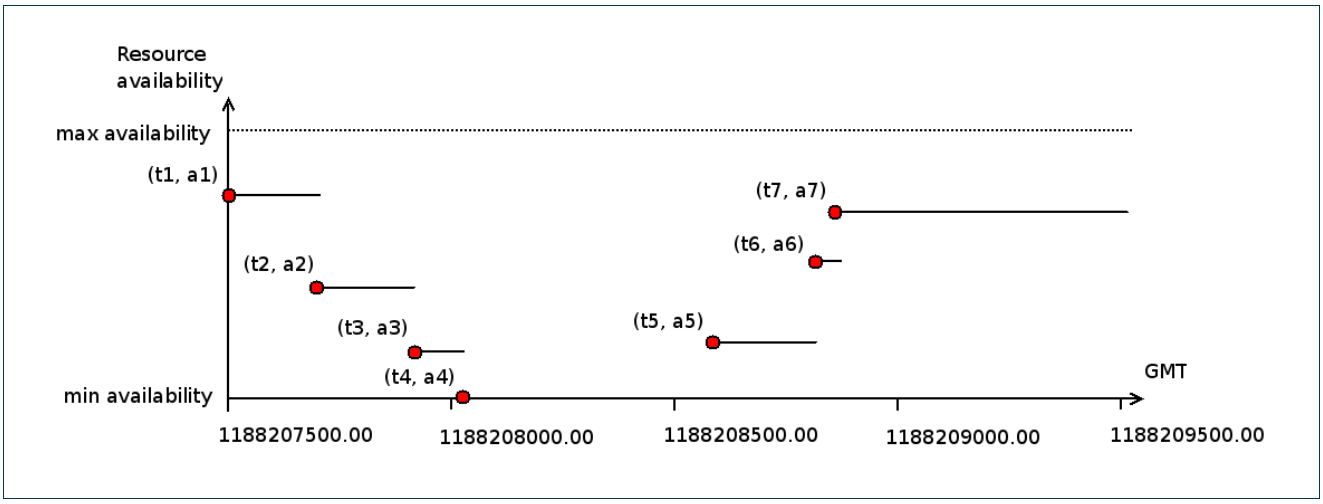
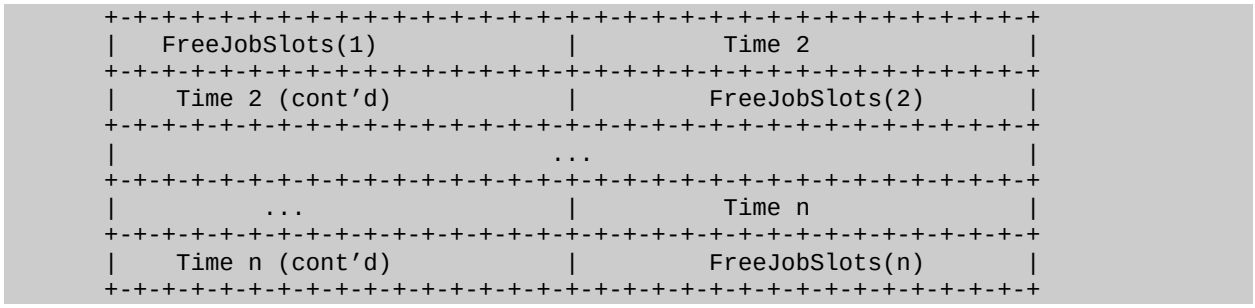


Figure 3-4: Resource availability calendar representation.

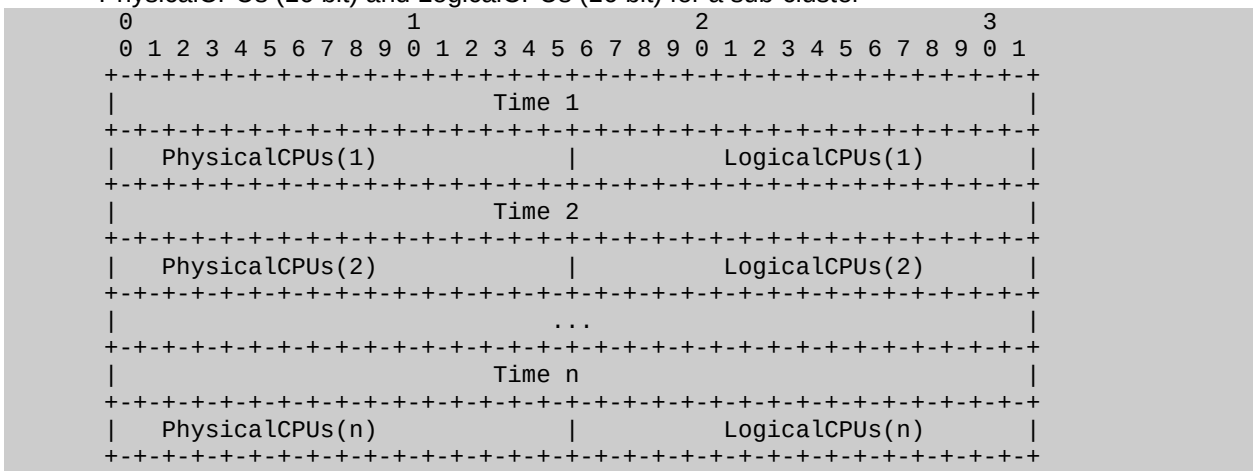
Depending on the type of resource (Grid or TE-link), the Availability field may contain different information. In details, it refers to:

- FreeJobSlots (16 bit) in case of a computing element

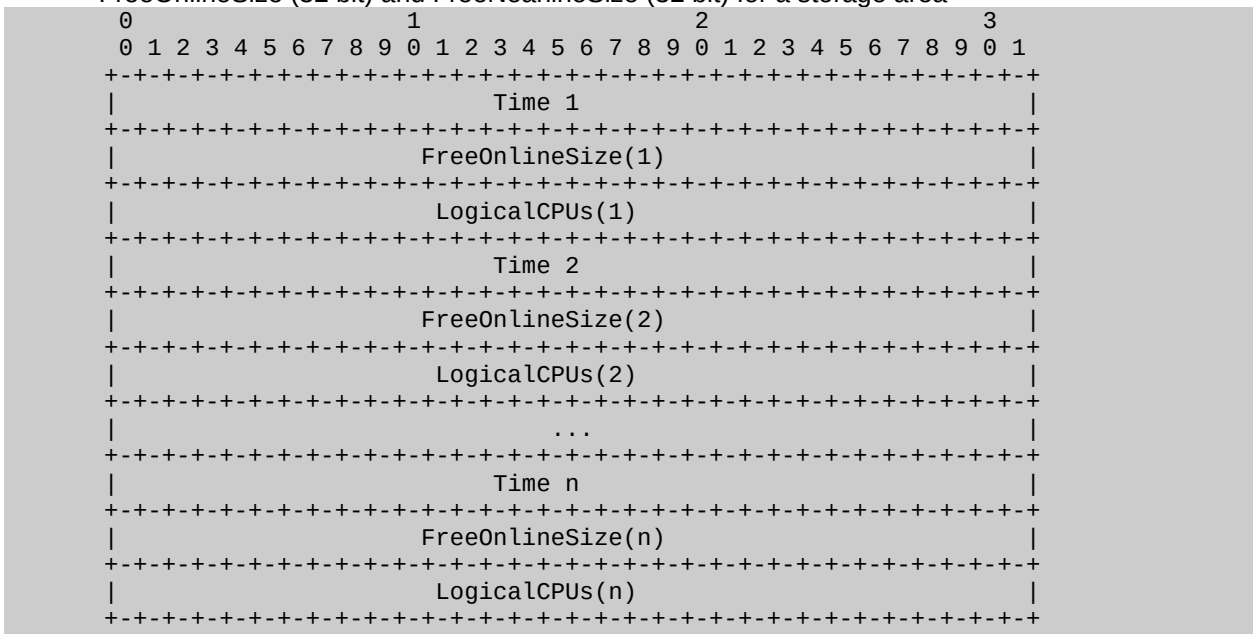
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Time 1                               |
```



- PhysicalCPUs (16 bit) and LogicalCPUs (16 bit) for a sub-cluster



- FreeOnlineSize (32 bit) and FreeNearlineSize (32 bit) for a storage area



- UnreservedBandwidth per prio (8* 32 bit) for a TE-link

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Time 1                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[0](1)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[7](1)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Time 2                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[0](2)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[7](2)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ...                             |
|                               ...                             |
|                               ...                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Time n                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[0](n)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unreserved bandwidth[7](n)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3 Full-optical TE routing extensions in G^{2}MPLS

With reference to the architectural and theoretical framework presented in [PH-WP2-D2.1] about optical impairments experimented by full-optical networks, a number of additional parameters may need to be added and flooded about full-optical TE-links for an enhanced constraint based path computation. Among those identified in deliverable D2.1 we select here:

- **D_{PMD} (32 bits),**
  - it is the fiber Polarization Mode Dependent parameter in ps per sqrt(km) of the k-th span (TE-link) in the circuit, in 32 bit IEEE floating point format
- **Span length (32 bits),**
  - it represents the total length of the all-optical span in meters
- **List of amplifiers**
  - each entry describing gain G (32 bits value, units in dB) and noise figure n_{SP} (32 bit IEEE floating point format)
- **List of available wavelengths**,
  - a bit-mask multiple of 32 bits, to accommodate different wavelength sets from CWDM to DWDM.

Generation of these TE extensions is optional and cannot hamper the operation of a GMPLS/G^{2}MPLS Control Plane. Most of the identified data on physical characterization of the all-optical connections are practically unknown to the Network Operators, and this makes difficult an assessment of the full-optical extensions in a G^{2}MPLS testbed environment.

3.4 G^{2}MPLS opaque extensions for G.I-NNI OSPF (G^2.OSPF-TE)

The Traffic Engineering extensions for OSPF protocol rely on the mechanism of Opaque LSAs [IETF-RFC2370], which carry opaque application-specific information in a general container with different scopes (link, area, AS).

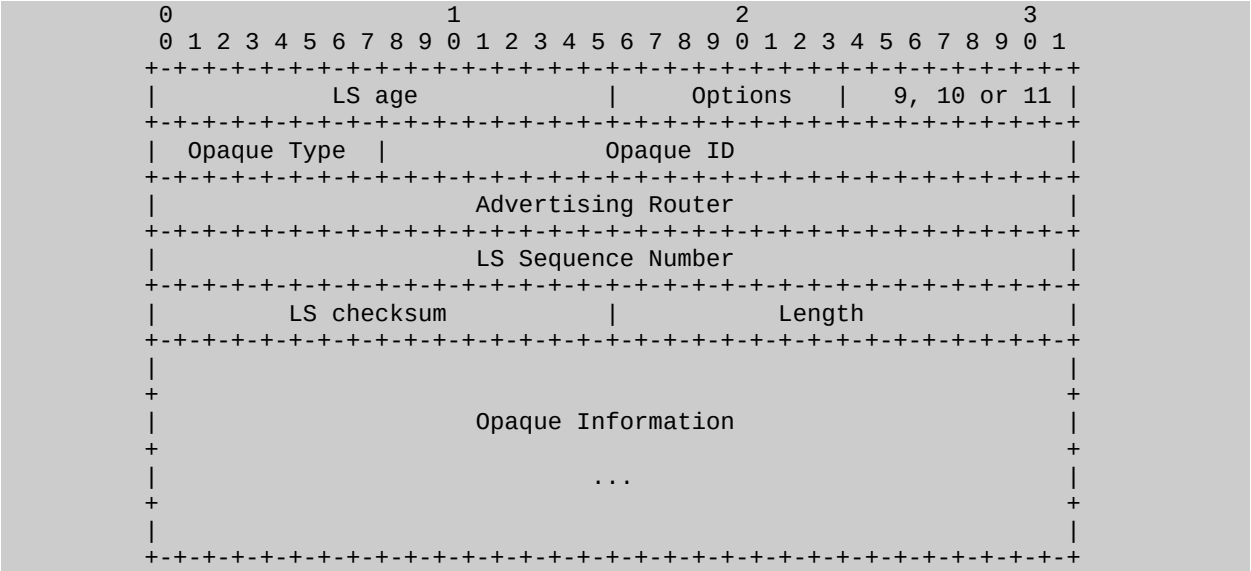


Figure 3-5: OSPF Opaque LSA.

The LSA ID (32 bits) of an Opaque LSA is split in two fields: 8 bits of type data and 24 bits of type-specific data. The new IANA policies outlined in [IETF-RFC4940] define the following assignment policies:

Opaque Type Range	Assignment Policy
0	*Not to be assigned*
1-3	*Already assigned*
4-127	*Standards Action*
128-247	*Reserved*
248-251	*Experimentation*
252-255	*Vendor Private Use*

Table 3-10: IANA policies for OSPF Opaque Type assignment.

The GMPLS Traffic Engineering LSA uses type 1. The remaining 24 bits are the Instance field, which is an arbitrary value used to maintain multiple Traffic Engineering LSAs. A maximum of 2^24 Traffic Engineering LSAs may be sourced by a single system. The LSA ID has no topological significance.

For G^{2}MPLS routing at the G.I-NNI, a new Opaque LSA is defined, i.e. the Grid LSA, using type 248 (experimental) and an Instance field with the same meaning of the TE LSA.

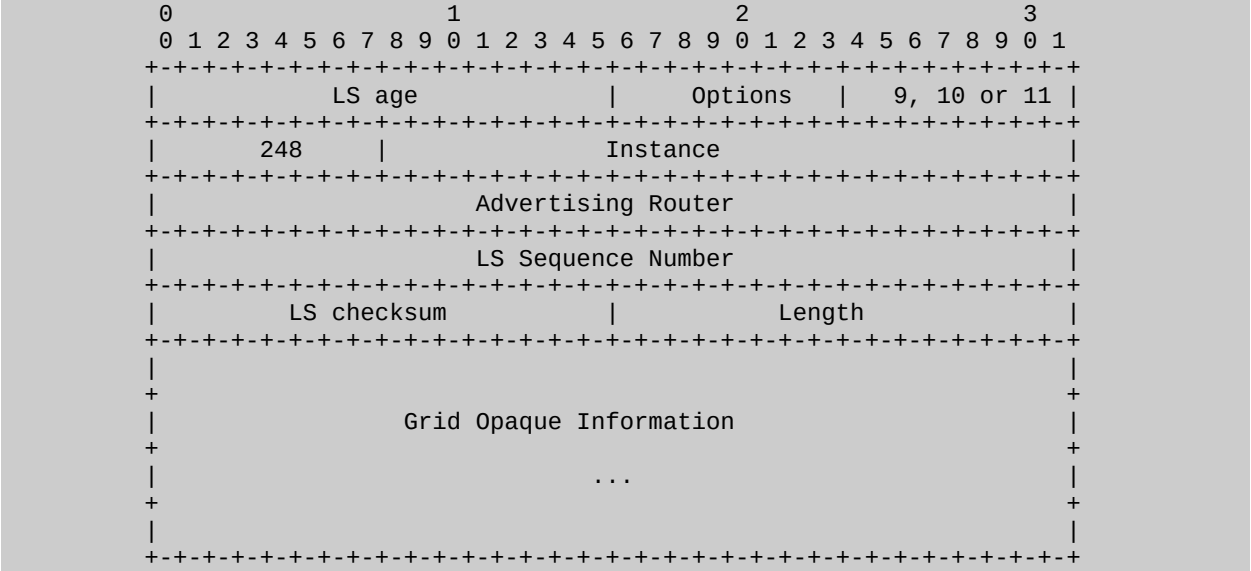


Figure 3-6: OSPF Grid Opaque LSA (type 248).

As per [IETF-RFC3630], the LSA payloads for TE LSA and Grid Site LSA consist of one or more nested Type/Length/Value (TLV) triplets, called the top-level TLV. Each top-level TLV may be further structured in sub-TLVs. Both have the format shown in Figure 3-7. The Length field defines the length of the value portion in bytes and TLV is padded to four-bytes alignment; padding is not included in the Length field.

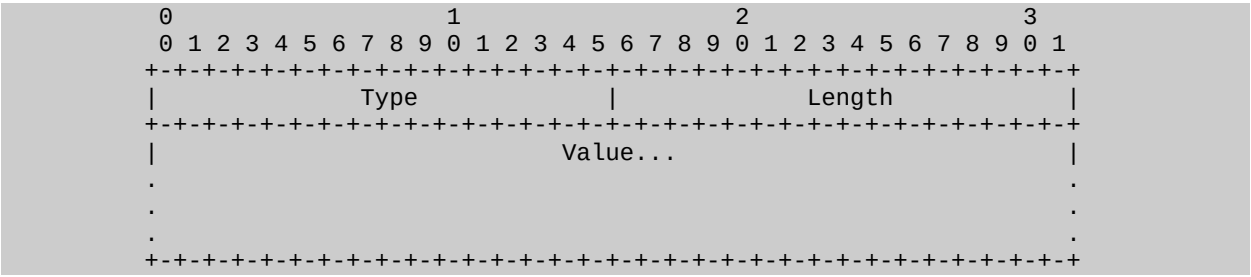


Figure 3-7: Top-level TLV format in OSPF Opaque LSA.

In [IETF-RFC 3630], [IETF-RFC 4203] and [OIF-E-NNI-Rtr-1.0] just one top-level TLV is contained in each TE LSA and three types of top-level TLVs are defined, as shown in Table 3-11. G^{2}MPLS extensions concerning resource availability calendars and full-optical extensions for TE-links are specified as additional and optional

sub-TLVs of the TE link top-TLV. On the contrary, Grid related extensions are completely new top- and sub-TLVs under the Grid opaque LSA. However, the rule about just one top-level TLV contained in each Grid LSA is maintained, for finer control of changes and traffic scalability purposes.

Top-TLV Type	Top-TLV Name	Description
1	Router address	It advertises a stable address of the advertising router in the Routing Area.
2	TE link	It advertise a TE-links in a given Routing Area
3	TNA address	It advertise TNAs in a given Routing Area

Table 3-11: TE LSA top level TLVs.

3.4.1 Grid LSA payload details

The top-level TLVs defined for the Grid LSA derive from GLUE schema mapping and are listed in Table 3-12.

Top-TLV Type	Top-TLV Name	Description
1	Grid Site	It describes a site with its location and name.
2	Grid Service	It describes an abstracted, logical view of actual software components.
3	Grid Computing Element	It describes the element that manages jobs and offers them execution environments.
4	Grid Sub-Cluster	It describes a homogeneous set of hosts.
5	Grid Storage Element	It describes an abstraction for a storage resource.

Table 3-12: Grid LSA top level TLVs.

The contents of these top-level TLVs are provided in the following sub-sections, and complement the descriptions and data types identified in section 3.1. Each top level and sub-level TLV has been assigned of temporary IANA type identifier in the experimental set.

3.4.1.1 *Grid Site TLV*

The Grid Site TLV describes a site with its location and name. It is constructed of a set of sub-TLVs. There are no ordering requirements for the sub-TLVs. Only one Grid Site TLV can be carried in each Grid LSA, allowing for fine granularity changes in the distributed information models.

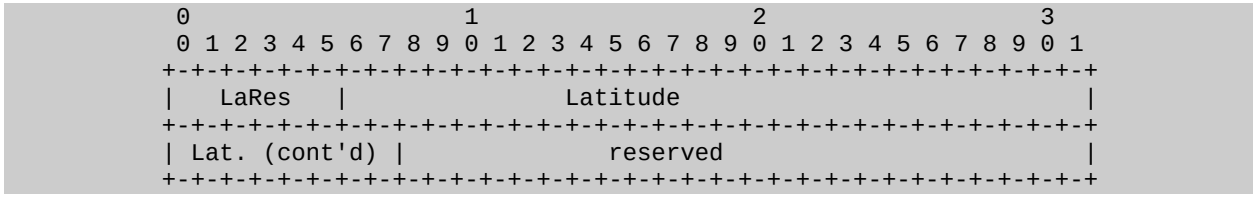
Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
1	4	ID	Unique Identifier of the Site	1	M

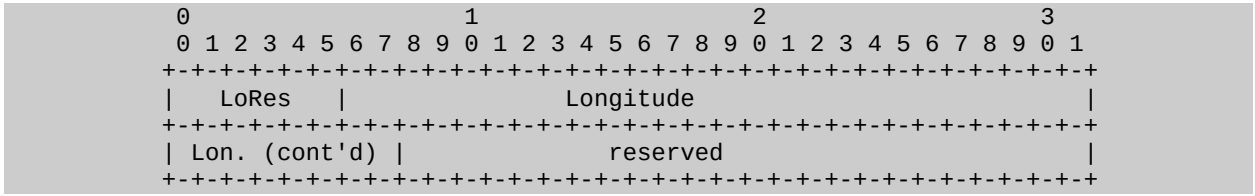
2	var	Name	Human-readable name	1	O
3	5	Latitude	Degree the position of a place north or south of the equator measured from -90° to +90° with positive values going north and negative values going south	1	O
4	5	Longitude	Degree the position of a place east or west of Greenwich, England measured from -180° to +180° with positive values going east and negative values going west	1	O

Table 3-13: Grid Site sub-TLVs.

Site ID (32 bits) is assigned during the address planning of the infrastructure, as well as the variable length Name string.

As per IETF RFC3825, latitude and longitude can be represented in fixed-point 2s-complement binary degrees, for the economy of a smaller option size compared to a string encoding of digits. The integer parts of these fields are 9 bits long to accommodate +/- 180 degrees. The fractional part is 25 bits long, better than the precision of 7 decimal digits. The length of each field is 40 bits, 34 of which is the sum of the integer (9) and fractional (25) bits, plus 6 bits of resolution.





3.4.1.2 *Grid Service TLV*

The Grid Service TLV describes a single abstracted service exported by the site. It is constructed of a set of sub-TLVs. There are no ordering requirements for the sub-TLVs. Only one Grid Service TLV can be carried in each Grid LSA, allowing for fine granularity changes in the distributed information models.

Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
1	4	ID	Identifier of the Service. ID is unique in the site	1	M

2	4	Parent Site ID	Identifier of the Grid Site that is exporting this service.	1	M
3	4	Service Info	The service info including service type and version	1	M
4	1	Status	Status of the service (OK, Warning, Critical, Unknown, Other)	1	O
5	1	Address Length	The Address length specifies the length of the endpoint address specified in number of bits.	1	M
6	4	IPv4 Endpoint	Network endpoint for this service. An Endpoint can appear just once for a Grid Service TLV.	1	M
7	16	IPv6 Endpoint		1	M
8	20	NSAP Endpoint		1	M

Table 3-14: Grid Service sub-TLVs.

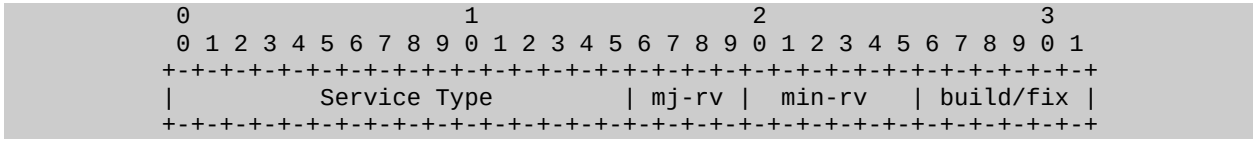
Service ID (32 bits) is an identifier local to the Grid site assigned for the service during the configuration phase of the site.

Parent Site ID (32 bits) is an identifier of the Grid site that is exporting the service.

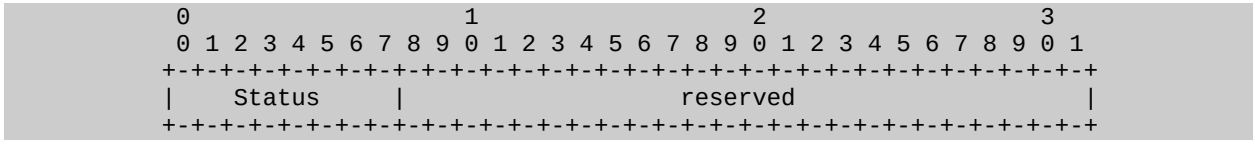
Endpoint is the address of the endpoint hosting this service and it is in the form of an IPv4 (32 bits), IPv6 (128 bits) or NSAP (160 bits) address.

The Address Length specifies the length of the endpoint address specified in number of bits. In single domain operations this needs to be the full length of endpoint address format, i.e. 32 bits (IPv4) or 128 bits (IPv6) or 160 bits (NSAP). In inter-domain operations this length could indicate sub-networks of homogeneous services under the same site.

Service Info has the following format, with Service Type taking a value in the grid_service_type_t and the version field reporting a hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits).



Status has the following format and can assume a value in the grid_service_status_t enumerated type.



3.4.1.3 *Grid Computing Element TLV*

The Grid Computing Element TLV describes computing resources available in the site. It is constructed of a set of sub-TLVs. There are no ordering requirements for the sub-TLVs. Only one Grid Computing Element TLV can be carried in each Grid LSA, allowing for fine granularity changes in the distributed information models.

Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
1	4	ID	Identifier of the Computing Element. ID is unique in the site	1	M
2	4	Parent Site ID	Identifier of the Grid Site that is exporting this computing element	1	M
3	4	LRMS Info	Type and version of the underlying local resource management system	1	M
4	1	Address Length	The Address length specifies the length of the host name address specified in number of bits.	1	M
5	4	IPv4 Host Name	Host name of the machine running this service. A hostname can appear just once for a Grid Computing Element TLV	1	M
6	16	IPv6 Host Name		1	M
7	20	NSAP Host Name		1	M
8	4	Gatekeeper Port	Gatekeeper port	1	O
9	var	Job Manager	The job manager used by the gatekeeper (e.g.: jobmanager-pbs). It distinguishes between different queues accessible using the same host and port.	1	O
10	var	Data Dir	DataDir is a POSIX compliant string representing the path of a run directory for jobs running on that Working Node.	1	O
11	4	Default Storage Element	the unique identifier of the default Storage Element for this computing element.	1	O
12	4	Jobs States	It contains the number of free job slots (sometimes called logical CPUs), and the queue status	1	O
13	12	Jobs Stats	It contains the number of jobs in running state, those in waiting state and those in any state.	1	O
14	8	Jobs Time Performances	It contains the estimated time to last for a new job from the acceptance to the start of its execution and the worst time from the job being accepted by the service to the start of its execution.	1	O
15	16	Jobs Time Policy	It contains the maximum wall clock time, the maximum obtainable wall clock time, the default maximum CPU time allowed to each job by the batch system and finally the maximum obtainable CPU time that can be granted to the job upon user request	1	O

16	17	Jobs Load Policy	It contains: • the maximum allowed number of jobs in the CE, • the maximum allowed number of jobs in running state • the maximum number of jobs that can be in waiting state • Number of slots for jobs to be in running state • The maximum number of slots which could be allocated to a single job • the jobs' priority • the pre-emption flag	1	O
17	var	CE calendar	It contains the jobs scheduling calendar reporting the available FreeJobsSlots (16 bit) for each timestamp.	1	O

Table 3-15: Grid Computing Element sub-TLVs.

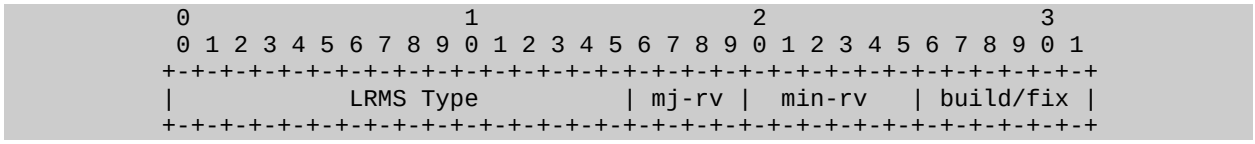
Computing Element ID (32 bits) is an identifier local to the Grid site assigned for the service during the configuration phase of the site.

Parent Site ID (32 bits) is an identifier of the Grid site that is exporting the computing element.

Hostname is the address of the machine running this service and it is in the form of an IPv4 (32 bits), IPv6 (128 bits) or NSAP (160 bits) address.

The Address Length specifies the length of the hostname address specified in number of bits. In single domain operations this needs to be the full length of endpoint address format, i.e. 32 bits (IPv4) or 128 bits (IPv6) or 160 bits (NSAP). In inter-domain operations this length could indicate sub-networks of homogeneous services under the same site.

LRMS Info has the following format, with LRMS Type taking a value in the grid_lrms_t and the version field reporting a hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits).

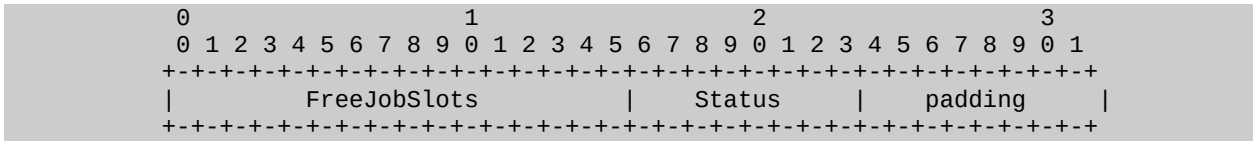


JobManager is a string identifying the job manager used by the gatekeeper (e.g.: jobmanager-pbs). Generally speaking, it is a string that distinguishes between different queues accessible using the same host and port.

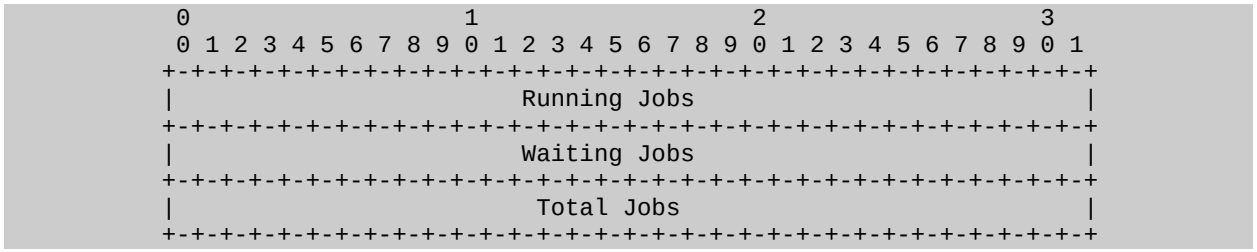
DataDir is a POSIX compliant string representing the path of a run directory for jobs running on that Working Node.

DefaultSE (32 bits) is the unique identifier of the default Storage Element for this computing element. Unique identifier of the default Storage Element to be used to store files from jobs in the CE in cases where no destination SE is explicitly stated.

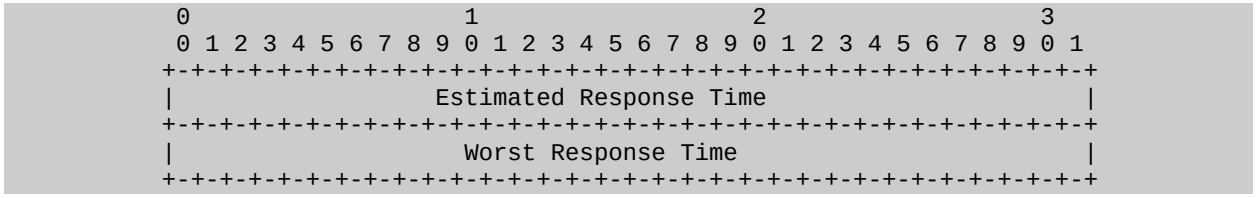
Jobs States contains the number of free job slots (sometimes called logical CPUs), and the queue status according to the following format:



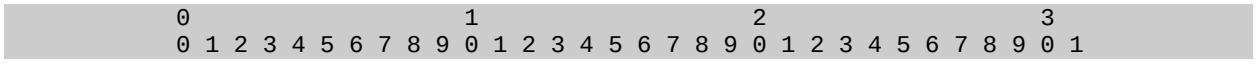
Jobs Stat, contains the number of jobs in running state, those in waiting state and those in any state according to the following format:

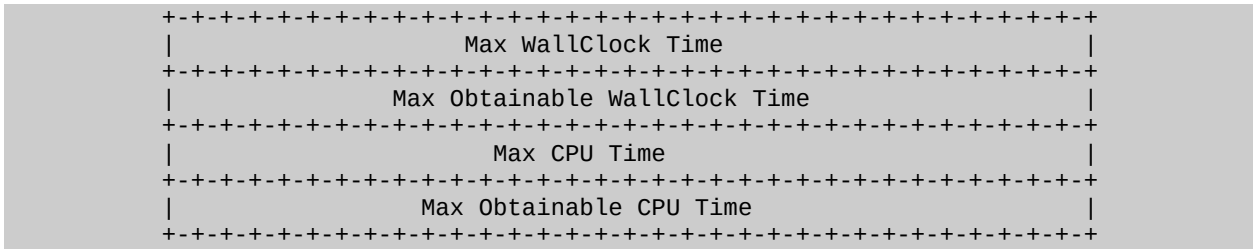


Jobs Time Performances contains the estimated time to last for a new job from the acceptance to the start of its execution and the worst time from the job being accepted by the service to the start of its execution. Sub-TLV format is:



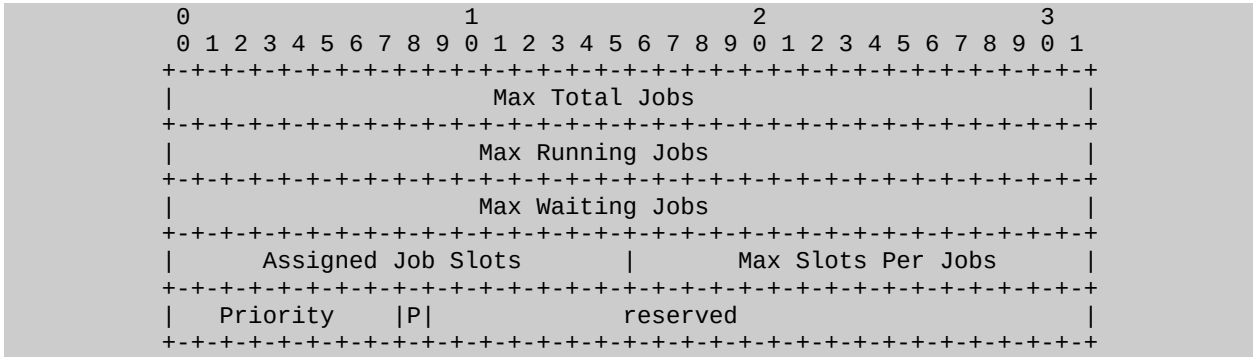
Jobs Time Policy contains the maximum wall clock time, the maximum obtainable wall clock time, the default maximum CPU time allowed to each job by the batch system and finally the maximum obtainable CPU time that can be granted to the job upon user request. Sub-TLV format is:



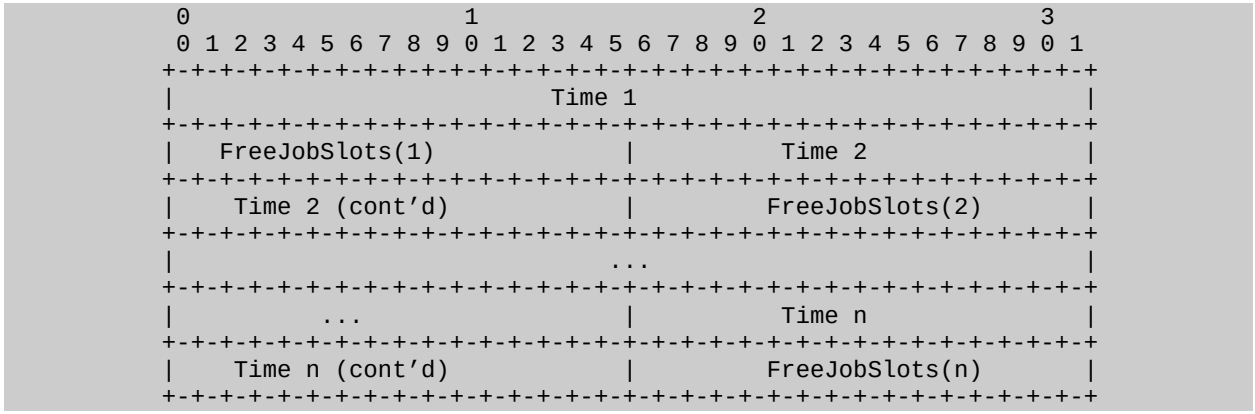


Jobs Load Policy has a longer TLV format and contains:

- the maximum allowed number of jobs in the CE,
- the maximum allowed number of jobs in running state
- the maximum number of jobs that can be in waiting state
- number of slots for jobs to be in running state
- the maximum number of slots which could be allocated to a single job
- the jobs' priority
- the pre-emption flag (P)



The CE calendar contains the jobs scheduling calendar in terms of the available FreeJobsSlots (16 bit) for each timestamp, according to the format:



3.4.1.4 *Grid Sub-Cluster TLV*

The Grid Sub-Cluster TLV describes a homogeneous set of hosts available in the site. It is constructed of a set of sub-TLVs. There are no ordering requirements for the sub-TLVs. Only one Grid Sub-Cluster TLV can be carried in each Grid LSA, allowing for fine granularity changes in the distributed information models.

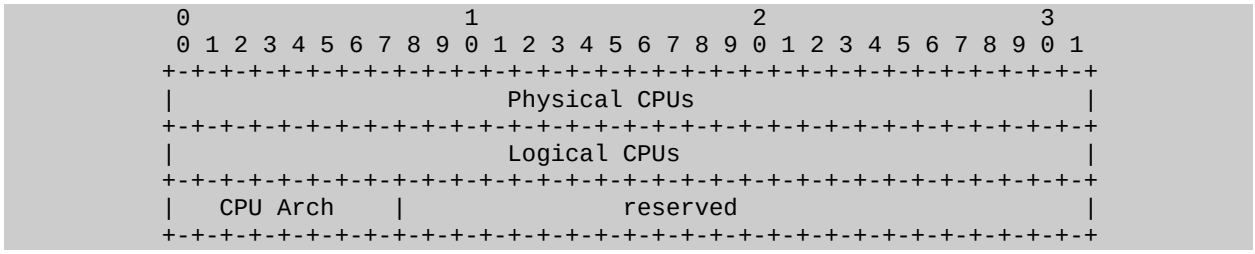
Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
1	4	ID	Identifier of the sub-cluster. ID is unique in the site.	1	M
2	4	Parent Site ID	Identifier of the Grid Site that is exporting this sub-cluster	1	M
3	9	CPU Info	It includes the CPU architecture, the total and the effective number of CPUs	1	O
4	4	OS Info	it contains information about the type of the OS and its version	1	O
5	8	Memory Info	it contains the amount of RAM and Virtual Memory, both in MB	1	O
6	var	Software Package	It contains information about an installed software package in terms of sw id, version and Environment Setup.	1…*	M
7	var	Sub-Cluster calendar	It contains the PhysicalCPUs (16 bit) and LogicalCPUs (16 bit) scheduling calendar for each timestamp	1	O

Table 3-16: Grid Sub-Cluster sub-TLVs.

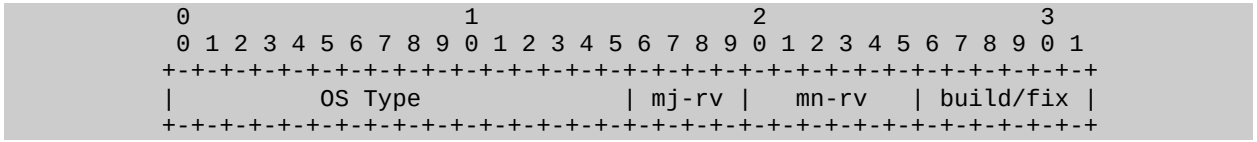
Sub-Cluster ID (32 bits) is an identifier local to the Grid site assigned for the service during the configuration phase of the site.

Parent Site ID (32 bits) is an identifier of the Grid site that is exporting the sub-cluster.

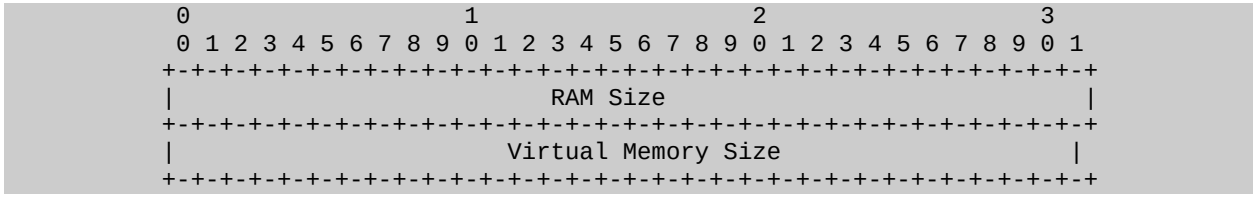
CPU Info describes the CPU architecture, derived from grid_cpu_arch_t, the total and the effective number of CPUs, according to the following format:



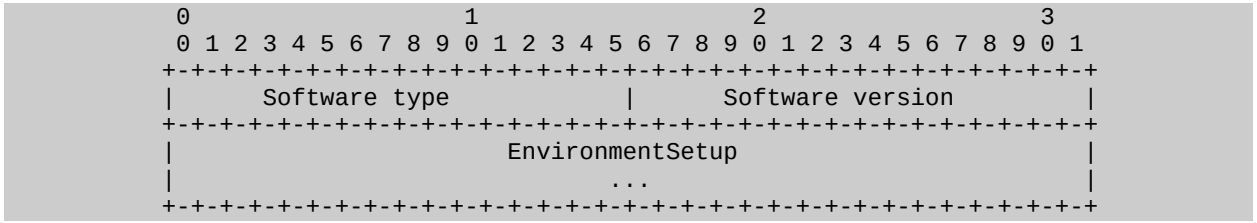
OS Info describes the type of the OS and its version deriving values from the enumerated grid_os_t (16 bits) and the 3 digits format of versions: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits). Sub-TLV format is:



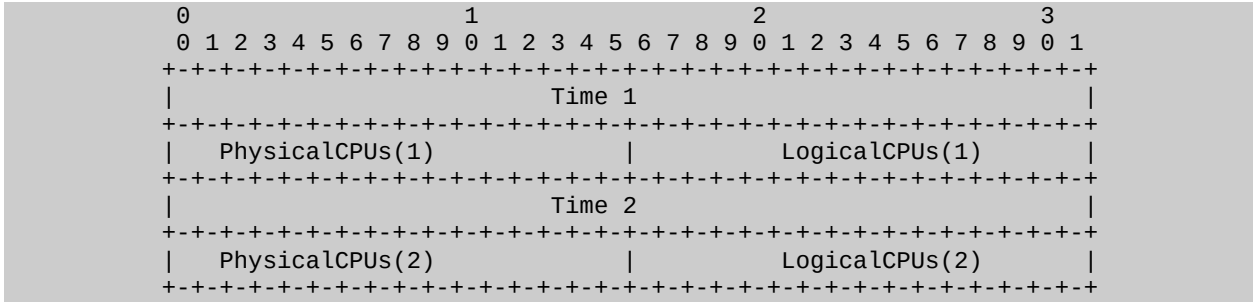
Memory Info reports the amount of RAM and Virtual Memory, both in MB, according to the following format:

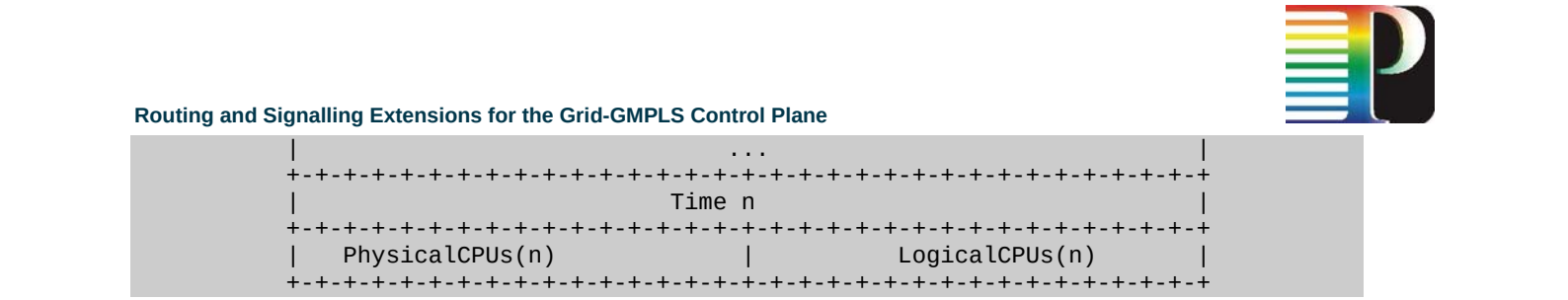


Each Software Package entry contains information about an installed software package. More than one entry may occur. Values for the software type derive from grid_software_t (16 bits) and versioning is always based on the 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits). EnvironmentSetup is a fully qualified sting for the setting of the application environment. Sub-TLV format is:



Sub-Cluster calendar contains the PhysicalCPUs (16 bit) and LogicalCPUs (16 bit) scheduling calendar for each timestamp. Sub-TLV format is:





```
|                                ...                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Time n                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   PhysicalCPUs(n)             |          LogicalCPUs(n)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.4.1.5 *Grid Storage Element TLV*

The Grid Storage Element TLV describes an abstraction for a storage resource, group of services, protocols and data sources available in the site. It is constructed of a set of sub-TLVs. There are no ordering requirements for the sub-TLVs. Only one Grid Storage Element TLV can be carried in each Grid LSA, allowing for fine granularity changes in the distributed information models.

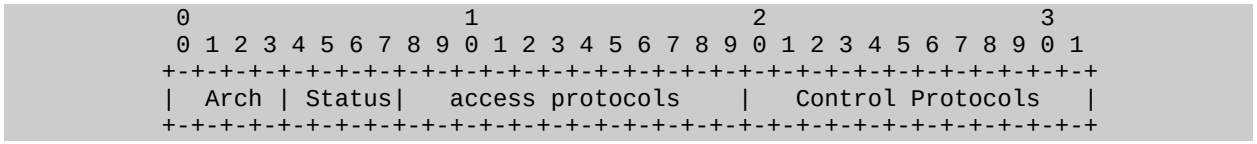
Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
1	4	ID	Identifier of the Storage Element. ID is unique in the site.	1	M
2	4	Parent Site ID	Identifier of the Grid Site that is exporting this storage	1	M
3	4	Storage Info	It contains information about the storage architecture, the status of the SE, the access and control protocols	1	M
4	8	Online size	It contains the online storage sizes (total + used) in GB	1	O
5	8	Nearline size	It contains the nearline storage sizes (total + used) in GB	1	O
6	var	Storage Area	It contains information about a portion of storage described in terms of name, path, sizes (total, used, reserved), management policies.	1…*	O
7	var	SE calendar	It contains the FreeOnlineSize (32 bit) and FreeNearlineSize (32 bit) scheduling calendar for each timestamp.	1	O

Table 3-17: Grid Storage Element sub-TLVs.

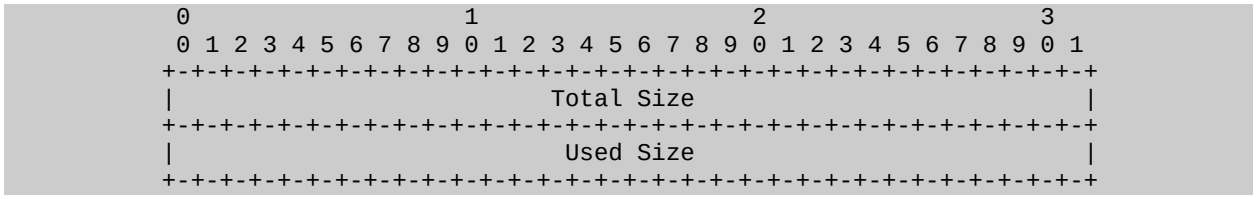
Storage Element ID (32 bits) is an identifier local to the Grid site assigned for the service during the configuration phase of the site.

Parent Site ID (32 bits) is an identifier of the Grid site that is exporting the storage.

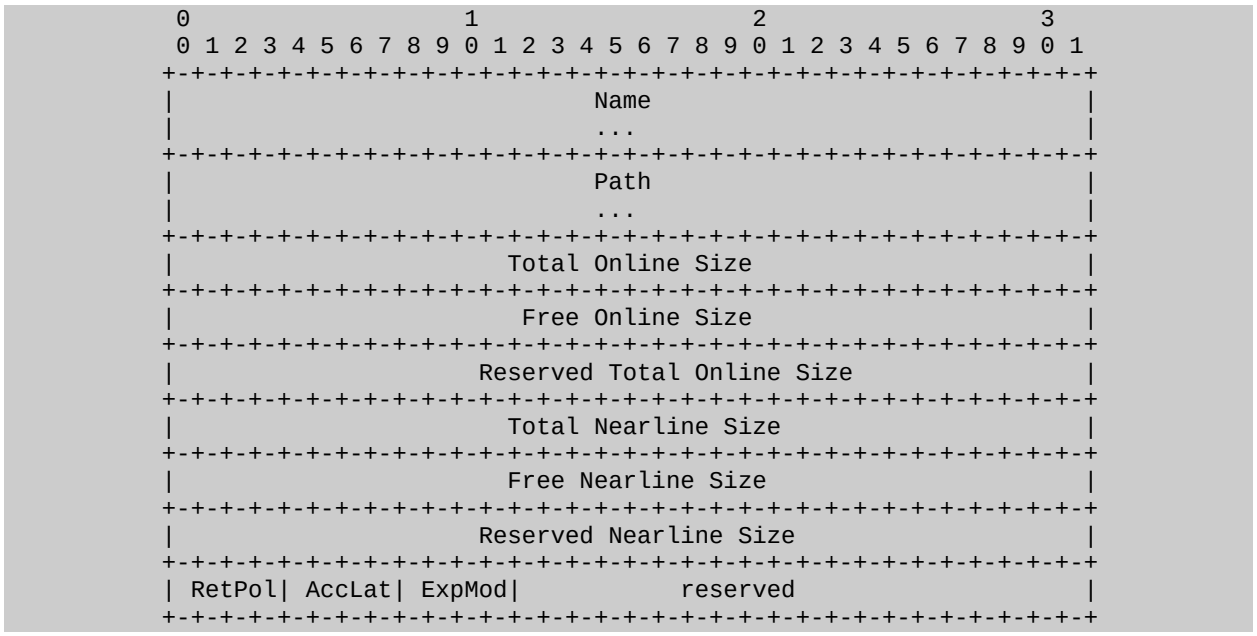
Storage Info describes the storage architecture by taking values from grid_storage_arch_t (4 bits), the status of the SE by taking values from grid_cese_status_t (4 bit), the access and control protocols in terms of bit-masks as specified in section 3.1.8. Sub-TLV format is:



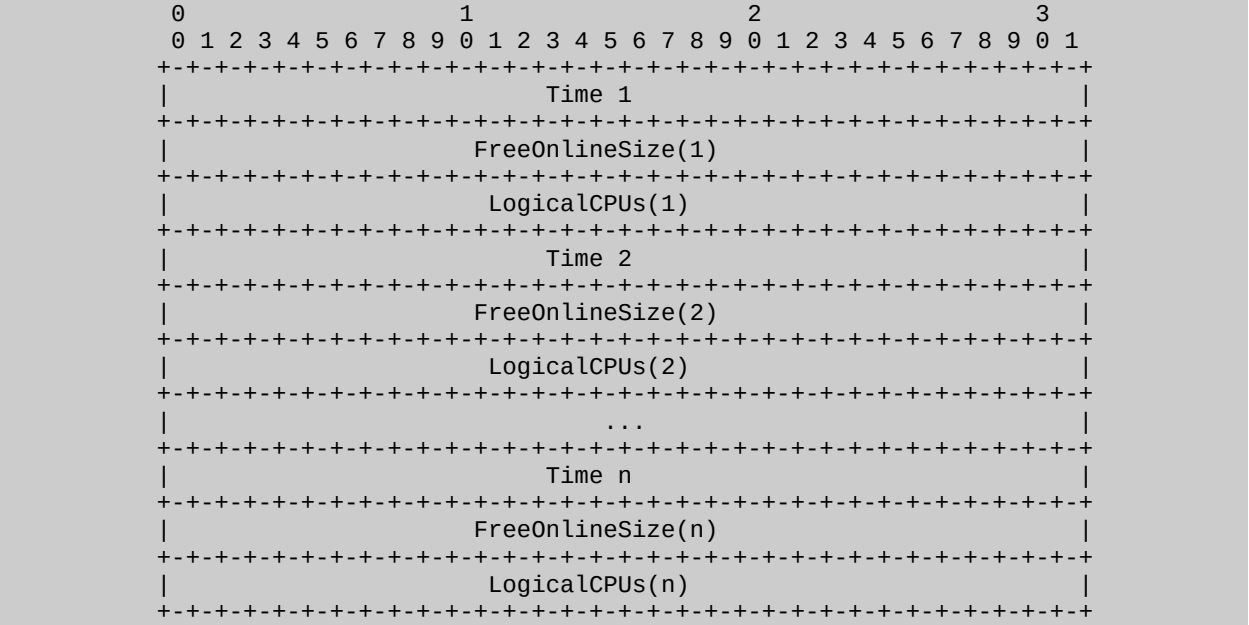
Online size and Nearline size report the amount of the respective used and total memories, both in GB, according to the following format:



Each Storage Area entry contains information about a portion of storage. More than one entry may occur. Names and path are variable length strings, the policy modes for the area derive their values from grid_retention_policy_t (4 bit), grid_access_latency_t (4 bit), grid_expiration_mode_t (4 bit). Memory sizes are in GB. Sub-TLV format is:



SE calendar contains the FreeOnlineSize (32 bit) and FreeNearlineSize (32 bit) scheduling calendar for each timestamp. Sub-TLV format is:



3.4.2 G²MPLS extensions to TE-Link top-level TLV

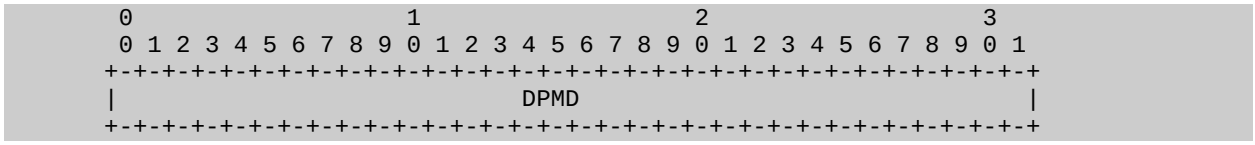
Additional sub-TLVs are defined for the full-optical modelling of TE-links. These are extensions to the standards base set for GMPLS TE-links, and have been assigned of temporary IANA type identifiers in the experimental set.

Sub-TLV Type	Sub-TLV Length	Sub-TLV Name	Description	Occurrence	Optionality
32780	4	DPMD	It is the fiber Polarization Mode Dependent parameter in ps per sqrt(km) of the k-th span (TE-link) in the circuit, in IEEE floating point format	1	O
32781	4	Span length	it represents the total length of the all-optical span in meters	1	O
32782	N* 4	List of amplifiers	each entry describing gain G (32 bits value in dB) and noise figure nSP (32 bit IEEE FP)	1	O
32783	N*4	List of available wavelengths	in the form of a bit-mask multiple of 32 bits, to accommodate different wavelength sets from CWDM to DWDM	1	O
32784	var	TE-link calendar	It contains scheduling calendar for each timestamp of the UnreservedBandwidth per priority (8* 32 bit)	1	O

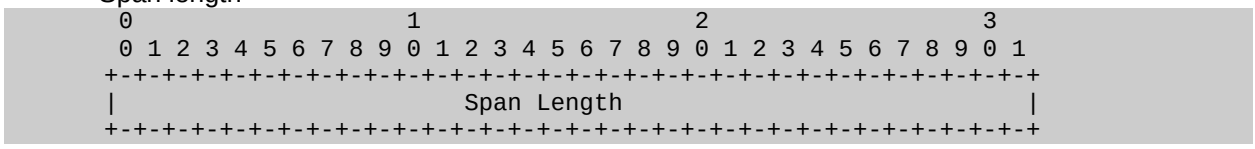
Table 3-18: TE-link additional sub-TLVs.

Sub-TLV formats result as follows:

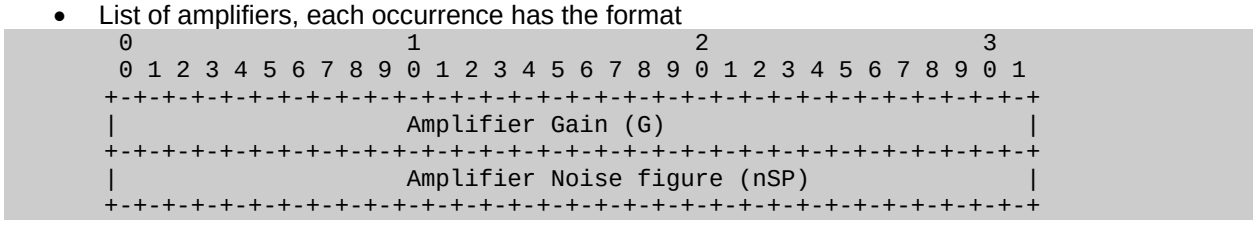
- DPMD



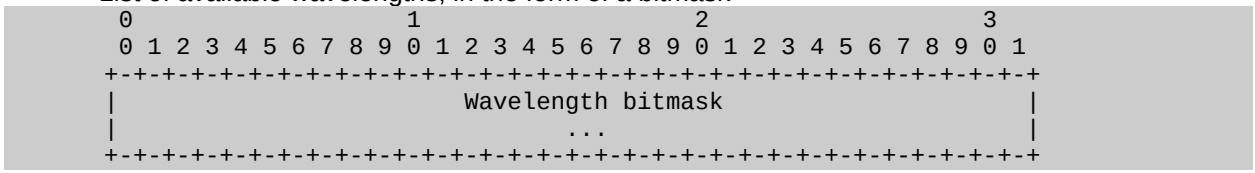
- Span length



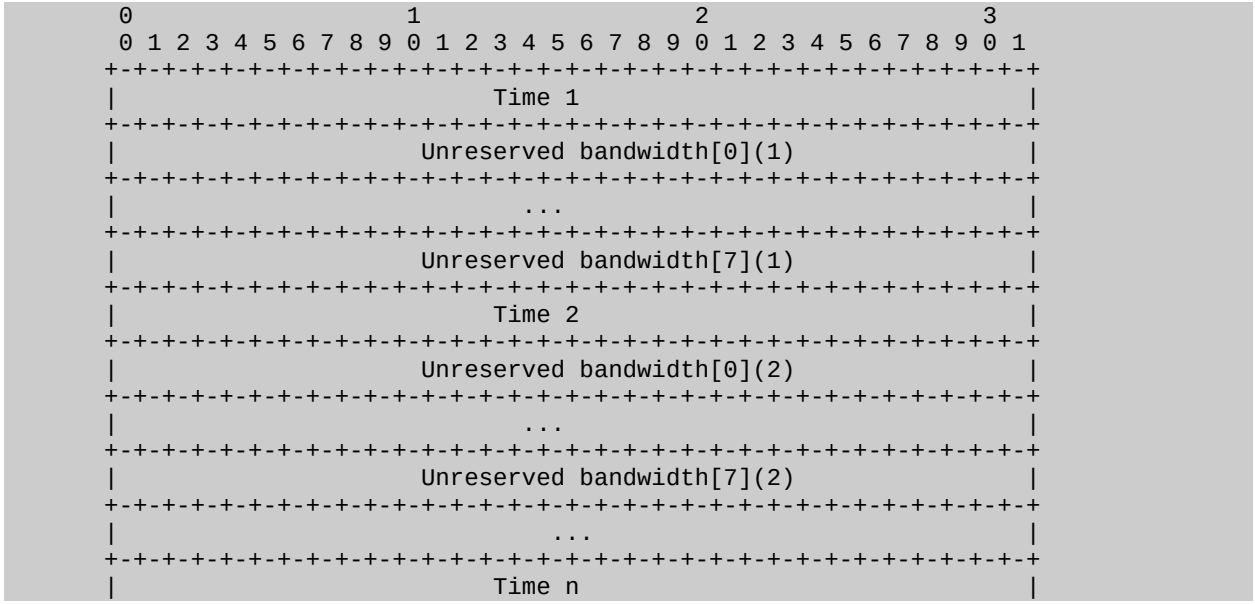
- List of amplifiers, each occurrence has the format



- List of available wavelengths, in the form of a bitmask



- TE-link calendar



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Unreserved bandwidth[0](n)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Unreserved bandwidth[7](n)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.5 G²MPLS opaque extensions for G.E-NNI OSPF (G².ENNI-OSPF)

G.E-NNI OSPF runs among Routing Controllers (RC), which are elected in each G²MPLS routing domain to implement a 2-layers routing hierarchy and establish adjacencies with (possibly not physically adjacent) peering RCs through the point-to-multipoint OSPF method.

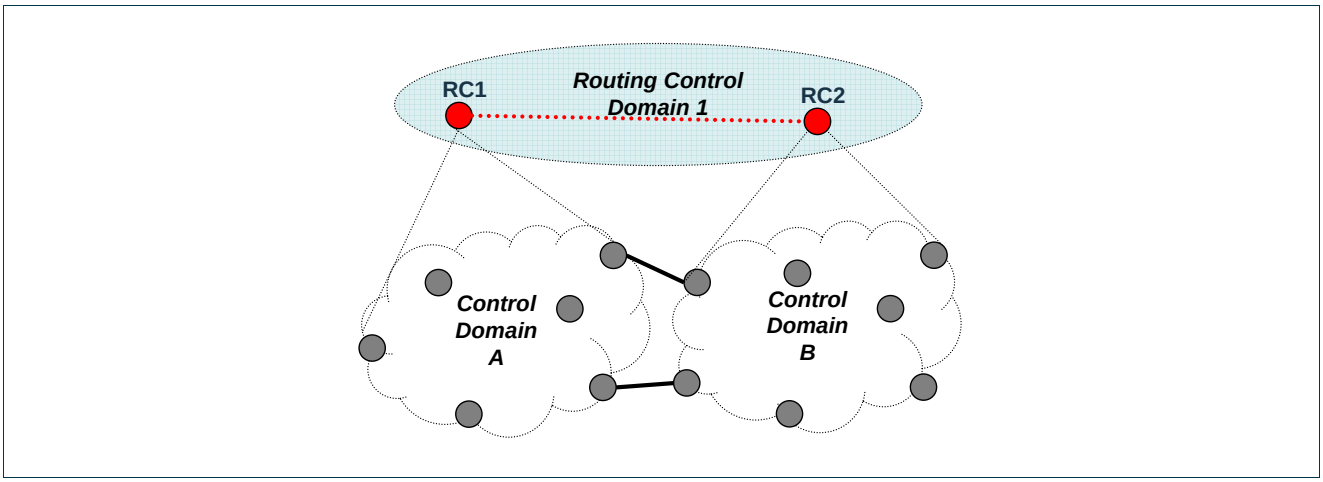


Figure 3-8: G.E-NNI 2-layers routing hierarchy.

G.E-NNI routing extensions are based on the standard OIF E-NNI OSPF extensions plus the G².OSPF-TE extensions identified in section 3.4.

G²MPLS extensions to TE-link top-level TLV must be considered optional parameters that can be added with the same TLVs identified in section 3.4 to the description of both inter-domain and intra-domain TE links. In G²MPLS routing, intra-domain links continue to be optional advertisements for a domain as in [OIF-E-NNI-Rtr-1.0].

Grid LSA top level TLVs identified in section 3.4 publish a mix of detailed and summarized information about a Grid site, which might result in scalability issues of the routing database and traffic. In particular,

- Grid Site TLV contains optional information that can be filtered at the G-E-NNI routing because the mandatory field (site ID) is contained in the other top-level TLVs to create the needed information model bindings.
- Grid Sub-Cluster and Storage Element TLVs natively summarize information related to the Grid site and it is a configuration policy to publish them in the form of the minimum homogeneous bundled set (e.g. same storage type or CPU architecture) both at the G-I-NNI and, consequently, at the G.E-NNI. Therefore, Grid Sub-Cluster and Storage Element TLVs are fed up and down unmodified by RCs, because any summarization would imply the removal of the site ID identifier and, consequently, the loss of localization information for those resources in the inter-domain advertisement.
- Grid Service and Computing Element TLVs represent very deep characterizations of some endpoints in a Grid site up to the detail of the full endpoint addresses (IPv4, IPv6 or NSAP). Therefore, these TLVs could/should be summarized during the feed-up at RCs with a bundling of homogeneous mandatory information (e.g. LRMS Info for Computing Element, Service Info for Service TLVs). Summarization will sum up resource availabilities, intersect calendars and group endpoint addresses through sub-netting techniques by means of the address length field in those TLVs.

4 G²MPLS Signalling Extensions

This chapter specifies the signalling extensions for Grid-enabled GMPLS.

Additional objects and capabilities identified here are intended to be optional additions to those defined for standard GMPLS signalling (IETF RFC3471 and IETF RFC3473-IETF RFC3209 for G.RSVP-TE), ASON DCM (ITU-T G.7713 and G.7713.2), and OIF UNI and E-NNI signalling (OIF UNI 1.0R2, ENNI-sig-01.0). For this reason, just Grid job modelling and extensions as well as special signalling procedures for Phosphorus (including resource calendars and full-optical TE-links) are presented in this chapter. Standard TE extensions for GMPLS and OIF UNI and ENNI signalling are not presented, though they are fundamental and integral part of any G²MPLS signalling implementation. Formatting and composition rules for the standard objects can be found in the related standards.

The selected signalling protocol for G²MPLS operations is G.RSVP-TE, with its standard extensions contributed by IETF for the GMPLS part and by OIF for the UNI and E-NNI parts.

G²MPLS extensions to RSVP-TE for the different G²MPLS network reference points (G.OUNI, G.I-NNI, G.E-NNI) are conceived to cope with:

- Grid job/service requests translated in setup of GNS transactions
- support of advance reservations
- support of implicit network destinations for Calls related to a Grid service (destination is an amount of CPU or storage wherever it is)

Integration with Grid AAA and Security infrastructures and escalation of recovery procedures from the network domain to the Grid middleware are not part of this extension set. These issues are planned for a later analysis and specification between WP2 and WP4 and WP3 respectively. Any further extensions to the signalling resulting from those activities and not just included in the standard ASON/GMPLS protocols will be reflected in an update revision of this document.

4.1 Grid job modelling in G^{2}MPLS

Job submission in Grids is based on a description language by which capabilities and amounts of resources (computational and/or storage) are specified among the involved sites.

A standardization effort for this fundamental phase of Grid operation is being carried out by OGF through the Job Submission Description Language working group (JSDL). JSDL is a standardized language for the submission of jobs among organizations using a variety of local job management systems. JSDL is currently available in version 1.0 with some minor revisions from the original GFD-0.56 specification (ref. [JSDL-1.0-r14]). JSDL is used as core vocabulary for describing job submissions by a number of existing operational systems such as: Condor, Globus Toolkit, Load Sharing Facility (LSF), Portable Batch System (PBS), (Sun) GridEngine (SGE), Uniform Interface to Computing Resources (Unicore).

According to the specification [JSDL-1.0-r14], each job is described by an XML document structured in information elements. The root element *JobDefinition* contains a single mandatory child element named *JobDescription*. The *JobDescription* contains further elements that describe the job: *JobIdentification*, *Application*, *Resources* and *DataStaging*. The JSDL Pseudo XML schema results as follows:

```
<JobDefinition>
        <JobDescription>
                <JobIdentification >
                        <JobName> xsd:string </JobName>?
                        <Description> xsd:string </Description>?
                        <JobAnnotation> xsd:string </JobAnnotation>*
                        <JobProject> xsd:string </JobProject>*
                </JobIdentification>

                <Application>
                        <ApplicationName> xsd:string </ApplicationName>?
                        <ApplicationVersion> xsd:string </ApplicationVersion>?
                        <Description> xsd:string </Description>?
                </Application>

                <Resources>
                        <CandidateHosts>
                                <HostName> xsd:string </HostName>+
                        </CandidateHosts>?
                        <FileSystem name="xsd:NCName">
                                <Description> xsd:string </Description>?
                                <MountPoint> xsd:string </MountPoint>?
                                <MountSource> xsd:string </MountSource>?
                                <DiskSpace> jsdl:RangeValue_Type </DiskSpace>?
                                <FileSystemType> jsdl:FileSystemTypeEnumeration </FileSystemType>?
                        </FileSystem>*
                        <ExclusiveExecution> xsd:boolean </ExclusiveExecution>?
                        <OperatingSystem>
                                <OperatingSystemType>
                                        <OperatingSystemName>
                                        jsdl:OperatingSystemTypeEnumeration
                                        </OperatingSystemName>
                                </OperatingSystemType>?
                                <OperatingSystemVersion> xsd:string </OperatingSystemVersion>?
                                <Description> xsd:string </Description>?
                        </OperatingSystem>?
                        <CPUArchitecture>
                                <CPUArchitectureName>
```

```
                              jsdl:ProcessorArchitectureEnumeration
                              </CPUArchitectureName>
                        </CPUArchitecture>?
                        <IndividualCPUSpeed> jsdl:RangeValue_Type </IndividualCPUSpeed>?
                        <IndividualCPUTime> jsdl:RangeValue_Type </IndividualCPUTime>?
                        <IndividualCPUCount> jsdl:RangeValue_Type </IndividualCPUCount>?
                        <IndividualNetworkBandwidth>
                        jsdl:RangeValue_Type
                        </IndividualNetworkBandwidth>?
                        <IndividualPhysicalMemory>
                        jsdl:RangeValue_Type
                        </IndividualPhysicalMemory>?
                        <IndividualVirtualMemory> jsdl:RangeValue_Type </IndividualVirtualMemory>?
                        <IndividualDiskSpace> jsdl:RangeValue_Type </IndividualDiskSpace>?
                        <TotalCPUTime> jsdl:RangeValue_Type </TotalCPUTime>?
                        <TotalCPUCount> jsdl:RangeValue_Type </TotalCPUCount>?
                        <TotalPhysicalMemory> jsdl:RangeValue_Type </TotalPhysicalMemory>?
                        <TotalVirtualMemory> jsdl:RangeValue_Type </TotalVirtualMemory>?
                        <TotalDiskSpace> jsdl:RangeValue_Type </TotalDiskSpace>?
                        <TotalResourceCount> jsdl:RangeValue_Type </TotalResourceCount>?
                </Resources>?

                <DataStaging name="xsd:NCName"?>
                        <FileName> xsd:string </FileName>
                        <FileSystemName> xsd:NCName </FileSystemName>?
                        <CreationFlag> jsdl:CreationFlagEnumeration </CreationFlag>
                        <DeleteOnTermination> xsd:boolean </DeleteOnTermination>?
                        <Source>
                                <URI> xsd:anyURI </URI>?
                        </Source>?
                        <Target>
                                <URI> xsd:anyURI </URI>?
                        </Target>?
                </DataStaging>*
        </JobDescription>
</JobDefinition>
```

Bold-faced tags represent the main job description data. Most of the element tags are complex types (i.e. contains attributes or other elements) and can appear in specific portions of the document. Two tags, i.e. `<Description>` and `<URI>`, can occur in several elements.

Most of the JSDL properties are useful for a Control Plane automatic operation, both in Phosphorus Overlay and Integrated models. In the first case, i.e. Phosphorus Overlay, Grid information is just piggybacked opaquely across the different reference points (i.e. G.OUNI, G.I-NNI, G.E-NNI) because of the Network Service scope of G^{2}MPLS operations. In the latter case, i.e. Phosphorus Integrated, Grid information is transparent at the G.OUNI and, possibly, also at the G.I-NNI and G.E-NNI in order to support crankback in case of anycast GNS requests.
In both cases just a relevant and minimum set of properties has been selected from the JSDL schema for G^{2}MPLS signalling, in order to provide a sufficient detail of information during the GNS setup phase. The hierarchical schema derived from JSDL is shown in Figure 4-1. A detailed enumeration of the selected properties is provided in the following sub-sections, as well as the identification of the relevant types and bit fields for G^{2}MPLS signalling extensions. The JDSL attributes selected for G^{2}MPLS purposes are highlighted in green.

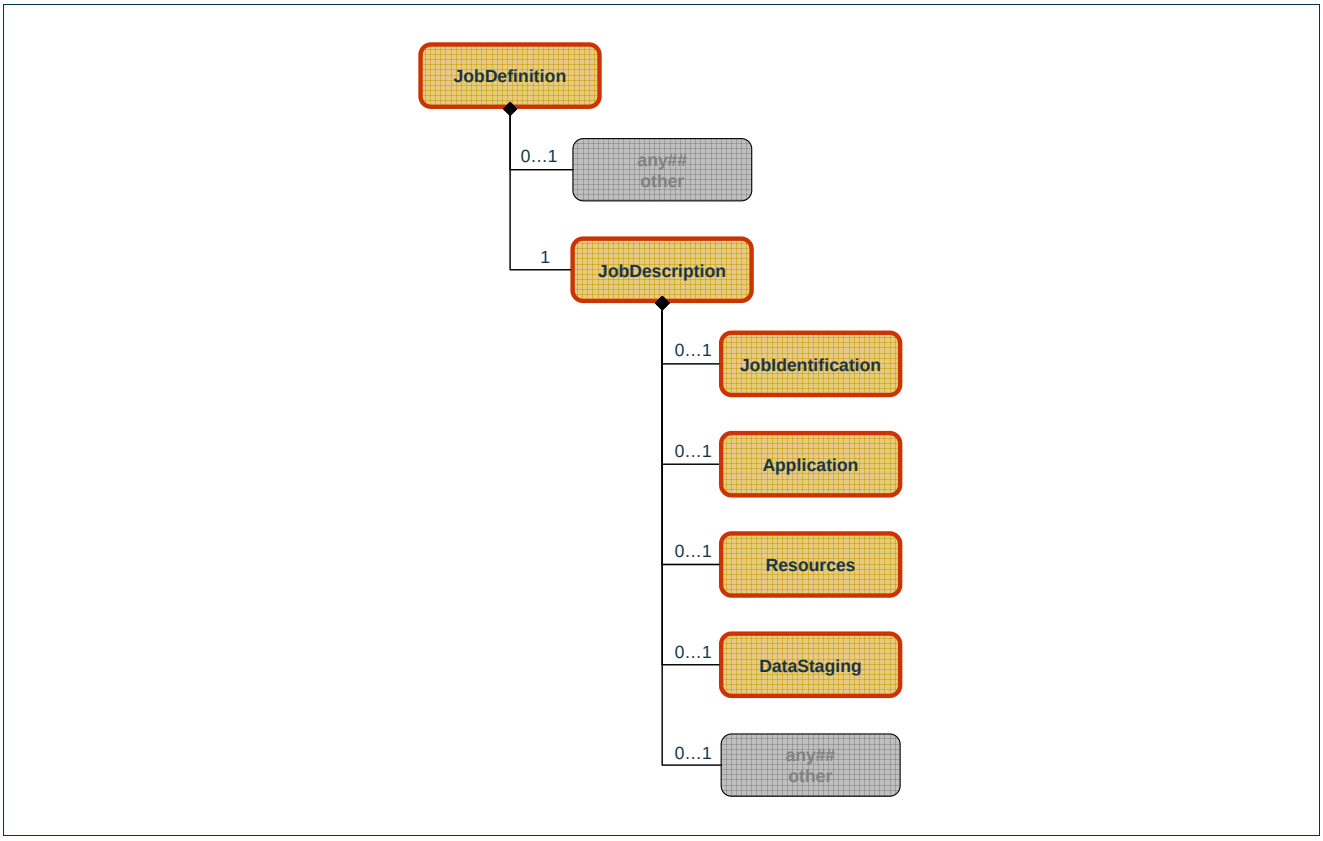


Figure 4-1: JSDL1.0 elements selection for G^{2}MPLS.

4.1.1 Job Definition

This element describes the job and its requirements. It contains a JobDescription section. It is the root element of the JSDL document.

JobDefinition schema/attributes	Description	Count	JSDL data type	G^{2}MPLS data type
ID	The ID of the job definition document. This is defined as a “xsd:ID” and is in the default namespace of the document. The ID MAY be omitted.	0…1	xsd::ID	64 bits
JobDescription	This element describes the job and its requirements.	1	COMPLEX TYPE	

Table 4-1: JSDL Job Definition and G^{2}MPLS selection of attributes.

For G^{2}MPLS just ID is selected. Job ID (64 bits) is a unique identifier for the job originated by the Grid middleware system. It remains constant over the life of the job and represents the GNS transaction identifier.

4.1.2 Job Description

This element describes the job and its requirements. It contains JobIdentification, Application, Resources, and DataStaging elements, which are complex types.

JobDescription schema/attributes	Description	Count	JSDL data type	G²MPLS data type
JobIdentity	This element contains all elements that identify the job: JobName, Description, JobAnnotation, and JobProject. If this element is not present then its value, including all of its sub-elements, is undefined.	0…1	COMPLEX TYPE	
Application	This element describes the Application and its requirements. It contains ApplicationName, ApplicationVersion and Description elements. It serves as a high level generic container that is intended to hold more specific application definitions. If this is not present then this job definition does not define an application to execute. The JSDL document could be defining a data staging job, or a null job.	0…1	COMPLEX TYPE	
Resources	This element contains the resource requirements of the job. If this element is not present then the consuming system MAY choose any set of resources to execute the job. Any combination of the listed Resources sub-elements MAY be present in the Resources element of a JSDL document. In particular, any combination of "Individual", and "Total" elements of the same or different types MAY be present in a Resources element.	0…1	COMPLEX TYPE	
DataStaging	Data staging defines the files that should be moved to the execution host (stage in) and the files that should be moved from the execution host (stage out). Files are staged in before the job starts executing. Files are staged out after the job terminates.	0…1	COMPLEX TYPE	

Table 4-2: JSDL Job Description and G²MPLS selection of attributes.

The component elements are part of the G²MPLS signalling extensions, but mapping is specified for the respective descriptive subsections.

4.1.3 Job Identity

This element contains all elements that identify the job: JobName, Description, JobAnnotation, and JobProject. If this element is not present then its value, including all of its sub-elements, is undefined.

Job Identity schema/attributes	Description	Count	JSDL data type	G²MPLS data type
JobName	This element is a string that MAY be specified by a user to name the job specified in the JSDL document. It may not be unique to a particular JSDL document, which means that a user MAY specify the same JobName for multiple JSDL documents. If this element is not present then it is not defined.	0…1	xsd::string	string
Description	This element provides descriptive, human readable, information about its containing complex element. It MAY be present as a sub-element of a number of other JSDL elements: Application, FileSystem, etc. If this element is not present as a sub-element then no description is defined.	0…1	xsd::string	
JobAnnotation	This element is DEPRECATED and should contain a string that MAY be specified by a user to annotate the job. If this element is not present then it is not defined. In contrast to the Description element, JobAnnotation MAY contain information that is intended for use by the consuming system.	0…1	xsd::string	
JobProject	This element is a string specifying the project to which the job belongs. The project could be used by accounting systems or access control systems. The interpretation of the JobProject elements is left to the implementation of the consuming system. If this element is not present then it is not defined.	0…1	xsd::string	string

Table 4-3: JSDL Job Identity and G²MPLS selection of attributes.

For the identification of a job, two optional elements are selected:

JobName (string), which is a string specified by the middleware (or by the user) to assign a human readable name to the job.

JobProject (string), which is a string specifying the project to which the job belongs.

If one or both elements are not present then they are not defined.

4.1.4 Job Application

This element describes the Application and its requirements. It uniformly describes an application by its name and version number. If this is not present then this job definition does not define an application to execute.

Application schema/attributes	Description	Count	JSDL data type	G²MPLS data type
ApplicationName	This element is a string that defines the name of the application and is used to identify the application independent of the location of its executable on a host or system. If this is not present then it is not defined and a null job is assumed unless there is an application extension element pre-sent that defines the application to execute.	0…1	xsd::string	grid_software_t (16 bits)
ApplicationVersion	This element is a string that defines the version of the application to execute. The consuming system MUST use exact textual match to select the version of the application. If this element is not present then it is not defined and any version of the application MAY be executed	0…1	xsd::string	version_t (16 bits)
Description	This element provides descriptive, human readable, information about its containing complex element. It MAY be present as a sub-element of a number of other JSDL elements: JobIdentification, FileSystem, etc. If this element is not present as a sub-element then no description is defined.	0…1	xsd::string	

Table 4-4: JSDL Application and G²MPLS selection of attributes.

For the identification of an application, two optional elements are selected.

ApplicationName (16 bits) is an identifier for the software name and can assume one of the following values in the grid_software_t :

- Wisdom 0x0001
- Kodavis 0x0002
- TOPS 0x0003
- DDSS 0x0004
- INCA 0x0005
- other 0xFFFF

ApplicationVersion (16 bits) provides the specific version of the application in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| mj-rv |   mn-rv   | build/fix |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.1.5 Job Resources

The resource element describes the resource requirements of the job. At most one Resources element may be used, grouping homogeneous resources. If this element is not present then the consuming system may choose any set of resources to execute the job.

Any combination of the listed Resources sub-elements may be present in the Resources element of a JSDL document. However, all elements present in a JSDL document must be satisfied for the entire document to be satisfied and “Individual” resource requirements cannot contradict “Totals” of the same type.

Resource schema/attributes	Description	Count	JSDL data type	G²MPLS data type
CandidateHosts	This element is a complex type specifying the set of named hosts which may be selected for running the job. If this element is present then one or more hosts from the set MUST be chosen to run the job. If this is not present then it is not defined. A named host may be a single host (e.g., a machine name), a logical group of hosts (e.g., a named logical group or cluster), a virtual machine, etc.	0…1	COMPLEX TYPE	IPv4, IPv6, NSAP address list
FileSystem	This element describes a filesystem that is required by the job. It is a complex type that may contain the location where the filesystem should be made available, the required amount of disk space and the type of the filesystem. The filesystem may be local to the resource (e.g., on a local disk), or may be remote (e.g., an NFS mount).	0…1	COMPLEX TYPE	see related details
ExclusiveExecution	This is a boolean that designates whether the job must have exclusive access to the resources allocated to it by the consuming system. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	xsd:boolean	1 bit
OperatingSystem	This is a complex type that defines the operating system required by the job. It may contain Description, OperatingSystemVersion, and OperatingSystemType elements. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	COMPLEX TYPE	see related details

Resource schema/attributes	Description	Count	JSDL data type	G²MPLS data type
CPUArchitecture	This element is a string specifying the CPU architecture required by the job in the execution environment. If this is not present then it is not defined and the consuming system MAY choose any value. Values not defined by the JSDL ProcessorArchitectureEnumeration MAY be used by specifying the special token “other” and including the value as an extension.	0…1	jsdl:ProcessorArchitectureEnumeration	grid_cpu_arch_t (8 bits)
IndividualCPUSpeed	This element is a range value specifying the speed of each CPU required by the job in the execution environment. The IndividualCPUSpeed is given in multiples of hertz. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits
IndividualCPUTime	This element is a range value specifying the total number of CPU seconds required on each resource to execute the job. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits
IndividualCPUCount	This element is a range value specifying the number of CPUs required on each individual re-source. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits
IndividualNetworkBandwidth	This element is a range value specifying the bandwidth requirements of each individual resource. The amount is specified as multiple of bits per second. If this is not present then it is not defined and the consuming system MAY choose any value. This element specifies a requirement on the maximum (nominal) bandwidth of a single network interface. For a resource to satisfy this requirement it MUST have at least one network interface (physical or otherwise) that can satisfy this requirement. There is no requirement, however, that a resource only has a single network interface; or that, if multiple interfaces are present, all the network interfaces on a resource match this requirement.	0…1	jsdl:RangeValue_Type	N * 32 bits
IndividualPhysicalMemory	This element is a range value specifying the amount of physical memory required on each individual resource. The amount is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits
IndividualVirtualMemory	This element is a range value specifying the amount of virtual memory required on each individual resource. The amount is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits

Resource schema/attributes	Description	Count	JSDL data type	G²MPLS data type
IndividualDiskSpace	This is a range value that describes the amount of disk space required on each individual re-source. The amount of disk space is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value. This element specifies a requirement on the maximum un-configured (raw) capacity of a disk on the resource. For a resource to satisfy this requirement it MUST have at least one disk that can satisfy this requirement. The disk may be a physical disk, a RAID volume, or some other type of block device. There is no requirement, however, that a resource only has a single disk; or that, if multiple disks are present, all the disks on a resource match this requirement.	0…1	jsdl:RangeValue_Type	N * 32 bits
TotalCPUTime	This element is a range value specifying total number of CPU seconds required, across all CPUs used to execute the job. If this is not present then it is not defined and the consuming system MAY choose any value. The consuming system SHOULD NOT terminate the job before the lower bound is reached.	0…1	jsdl:RangeValue_Type	N * 32 bits
TotalCPUCount	This element is a range value specifying the total number of CPUs required for this job submission. If this is not present then it is not defined and the consuming system MAY choose any value. This element is related to IndividualCPUCount.	0…1	jsdl:RangeValue_Type	N * 32 bits
TotalPhysicalMemory	This element is a range value specifying the required amount of physical memory for the entire job across all resources. The amount is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value. This element is related to IndividualPhysicalMemory.	0…1	jsdl:RangeValue_Type	N * 32 bits
TotalVirtualMemory	This element is a range value specifying the required total amount of virtual memory for the entire job across all resources. The amount is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value. This element is related to IndividualVirtualMemory.	0…1	jsdl:RangeValue_Type	N * 32 bits

Resource schema/attributes	Description	Count	JSDL data type	G²MPLS data type
TotalDiskSpace	This is a range value that describes the total amount of disk space required by the job across all resources. The amount of disk space is given in bytes. If this is not present then it is not defined and the consuming system MAY choose any value. This element is related to IndividualDiskSpace.	0…1	jsdl:RangeValue_Type	N * 32 bits
TotalResourceCount	This element is a range value specifying the total number of resources required by the job. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:RangeValue_Type	N * 32 bits

Table 4-5: JSDL Resource and G²MPLS selection of attributes.

CandidateHosts is a list of network addresses (IPv4, IPv6, NSAP) that may be selected for running the job. A named host may be a single host (e.g., a machine address), or a logical group of hosts (e.g., a cluster as defined for routing). If this element is present then one or more hosts from the set must be strictly chosen to run the job. If this is not present then it is not defined.

FileSystem is a complex type that describes a filesystem that is required by the job. Its attributes include:

- name of the filesystem (8 bits) with values from the g2mpls_FSName_t:
  - home 0x01
  - root 0x02
  - scratch 0x03
  - tmp 0x04
  - other 0xFF
- type of the filesystem (8 bits) with values from the grid_fs_type_t (8 bits)
  - swap 0x01
  - temporary 0x02
  - spool 0x03
  - normal 0x04
  - other 0xFF
- MountPoint, which is an optional string identifying the local mount point of the filesystem
- MountSource, which is an optional string identifying the remote (NFS) mount point of the filesystem
- DiskSpace, which is a variable length sequence of 32 bits integers used to identify the range of free disk space required by the job.

ExclusiveExecution (1 bit) is a flag indicating whether the job must have exclusive access to the resources allocated to it by the consuming system.

FileSystem schema/attributes	Description	Count	JSDL data type	G²MPLS data type
Name	The name of the FileSystem.	0…1	xsd:NCName	grid_fs_name_t (8 bits)
FileSystemType	This is a token that describes the type of filesystem of the containing FileSystem element. If this is not present then it is not defined and the consuming system MAY choose any value.	0…1	jsdl:FileSystemTypeEnumeration	grid_fs_type_t (8 bits)
MountPoint	This is a string that describes a local location that MUST be made available in the allocated re-sources for the job. If MountPoint is not defined the consuming system MUST choose the local location in which to provide the requested FileSystem.	0…1	xsd::string	string
MountSource	This is a string that describes a remote location that MUST be made available locally for the job (e.g. in NFS).	0…1	xsd::string	string
DiskSpace	This is a range value that describes the required amount of free disk space on the filesystem defined by the containing FileSystem element for the job.	0…1	jsdl:RangeValue_Type	N*32 bits

Table 4-6: JSDL FileSystem and G²MPLS selection of attributes.

OperatingSystem is a complex type that describes the operating system required by the job. Its attributes include:

- OperatingSystemType (16 bits), which can assume one of the values in the grid_os_t defined in section 3.1.7;
- OperatingSystemVersion (16 bits), which provides the specific version of an OS in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| mj-rv |   mn-rv   | build/fix |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

OperatingSystem schema/attributes	Description	Count	JSDL data type	G²MPLS data type
OperatingSystemType	This is a complex type that contains the name of the operating system. If this is not present then it is not defined and the consuming system MAY choose any value. Values not defined by the JSDL OperatingSystemTypeEnumeration may be used by specifying the special token “other” and including the value as an extension.	0…1	jsdl:OperatingSystemTypeEnumeration	grid_os_t (16 bits)

OperatingSystem schema/attributes	Description	Count	JSDL data type	G²MPLS data type
OperatingSystemVersion	This element is a string that defines the version of the operating system required by the job. The consuming system MUST use exact textual match to select the version of the operating system. If this is not present then any version of the operating system MAY be used.	0…1	xsd:string	version_t (16 bits)
Description	This element provides descriptive, human readable, information about its containing complex element. It MAY be present as a sub-element of a number of other JSDL elements: JobIdentification, FileSystem, etc. If this element is not present as a sub-element then no description is defined.	0…1	xsd::string	

Table 4-7: JSDL OperatingSystem and G²MPLS selection of attributes.

CPUArchitecture (8 bits) defines the CPU architecture required by the job in the execution environment and can assume values in grid_cpu_arch_t (8 bits) as defined in section 3.1.7. If this is not present then it is not defined and the consuming system may choose any value.

IndividualCPUSpeed, which is a variable length sequence of 32 bits integers used to identify the range of speeds required to each CPU for the execution of the job. Unit is Hertz. If this is not present then it is not defined and the consuming system may choose any value.

IndividualCPUTime, which is a variable length sequence of 32 bits integers used to identify the range of total number of CPU seconds required to each CPU for the execution of the job. Unit is seconds. If this is not present then it is not defined and the consuming system may choose any value.

IndividualCPUCount, which is a variable length sequence of 16 bits words used to identify the range of total CPUs required for the execution of the job. If this is not present then it is not defined and the consuming system may choose any value.

IndividualNetworkBandwidth, which is a variable length sequence of 32 bits integers in IEEE floating point format specifying the bandwidth requirements of each individual resource. Unit is bits per second. If this is not present then it is not defined and the consuming system may choose any value.

IndividualPhysicalMemory, which is a variable length sequence of 32 bits integers used to identify the range of physical memory required for the execution of the job. Unit is bytes. If this is not present then it is not defined and the consuming system may choose any value.

IndividualVirtualMemory, which is a variable length sequence of 32 bits integers used to identify the range of virtual memory required for the execution of the job. Unit is bytes. If this is not present then it is not defined and the consuming system may choose any value.

IndividualDiskSpace, which is a variable length sequence of 32 bits integers used to identify the range of disk space required for the execution of the job. Unit is bytes. If this is not present then it is not defined and the consuming system may choose any value.

TotalCPUTime, TotalCPUCount, TotalPhysicalMemory, TotalVirtualMemory, TotalDiskSpace and TotalResourceCount have the same format of the Individual fields but a scope related to the total count of resources required for job execution. Totals need to include Individuals values for consistency purposes.

4.1.6 Job DataStaging description

Data staging defines the files that should be moved to the execution host (stage in) and the files that should be moved from the execution host (stage out). Files are staged in before the job starts executing. Files are staged out after the job terminates.

DataStaging schema/attributes	Description	Count	JSDL data type	G²MPLS data type
Name	an optional name for the DataStaging element.	0…1	xsd:NCName	
FileName	This element is a string specifying the local name of the file (or directory) on the execution host. The FileName MUST be a relative path.	0…1	xsd::string	string
FilesystemName	If the FilesystemName is specified then the FileName is relative to the specified FileSystem declaration referenced by the name. In this case there MUST also be a FileSystem element with the same name.	0…1	xsd:NCName	grid_fs_name_t (8 bits)
CreationFlag	This element determines whether the file created on the local execution system can overwrite or append to an existing file. A typical value for this element, expected to be commonly supported, is “overwrite.”	0…1	jsdl:CreationFlagEnumeration	grid_ds_action_t (3 bits)
DeleteOnTermination	This is a boolean that determines whether the file should be deleted after the job terminates. If true the file is deleted after the job terminates or after the file has been staged out. Otherwise the file remains, subject to the persistency of the FileSystem it is on. If not present, behaviour is unspecified and depends on the consuming system.	0…1	jsdl:RangeValue_Type	1 bit
Source	A Source element contains the location of the file or directory on the remote system. This file or directory MUST be staged in from the location specified by the (optional) URI before the job has started. If this element is not present then the file does not have to be staged in.	0…1	uri	string

DataStaging schema/attributes	Description	Count	JSDL data type	G²MPLS data type
Target	A Target element contains the location of the file or directory on the remote system. This file or directory MUST be staged out to the location specified by the (optional) URI after the job has terminated. If this element is not present then the file or directory does not have to be staged out.	0…1	uri	string

Table 4-8: JSDL Data Staging and G²MPLS selection of attributes.

FileName is a string identifying a relative path in the execution host for a local file or directory to be used for staging.

FilesystemName is an optional field identifying the name of the filesytem in which the data staging file exists or will exist.

CreationFlag (3 bits) is a field used to specify the action to be performed on the file staged in or out with values from the g2mpls_DSAction:

- overwrite 0b001
- append 0b010
- dontOverwrite 0b100

DeleteOnTermination is a flag that determines whether the file should be deleted after the job terminates. If true the file is deleted after the job terminates or after the file has been staged out. Default value depends on the consuming system but it can be assumed false.

Source is an URI compliant with IETF RFC3986 specification identifying the location (host) in which the file or directory to be staged in can be found. Stage in occurs before job starting. If this element is not present then the file does not have to be staged in.

Target is an URI compliant with IETF RFC3986 specification identifying the location (host) in which the file or directory must be staged out. Stage in occurs after job end. If this element is not present then the file does not have to be staged out.

4.2 Special signalling procedures

4.2.1 GNS transaction setup

GNS transaction is the logical container that provides a common root to different network call/connections traversing the same G^{2}MPLS User to Network reference point. It is related to the Job ID, which is a unique identifier for the job originated by the Grid middleware system (or by the user) that remains constant over the life of the job. GNS transaction is shareable in a distributed way in the G^{2}MPLS NCP, in order to enable different invocation models:

- *direct invocation model*, in which user is co-located at the same Grid site of the some Grid resources be involved in the job and issues a GNS transaction creation including also remote Grid sites;
- *indirect invocation model*, in which user is located remotely with respect to all the Grid sites to be involved in the job execution and issues a remote GNS transaction creation between them.

For G^{2}MPLS purposes a job to be executed among Grid sites and specified through JSDL can be decomposed into different job segments (parts) each with its own description of the requested/involved resources.

Each job segment is served by a G^{2}MPLS call, which is an extension of the ASON call[2] with reference to the parent GNS transaction and temporal specifications of the service.

The introduction of the GNS transaction concept enables 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. For example, in a distributed computing and visualization application (e.g. Kodavis) it might be possible to have one connection between a data storage resource/location and a computing resource/location and another connection between the computing resource/location and a visualization client. Similar examples can be described for distributed storage experiments.

The submission of a job results in the setup of a GNS transaction, which in turn results in the signalling of a number of calls between the endpoints, being them explicitly selected like in standard Network Services or implicitly inferred by the resource characterization (anycast request). Signalling flows for the different phases across the different network reference points (G.OUNI, G.I-NNI, G.E-NNI) are similar to those defined in [ASON-DCM] and [ASON-RSVP]. They map the abstract messaging identified [PH-WP2-D2.1] for these network interfaces into RSVP messages (ref. Figure 4-2). Details on the different phases are in the scope of deliverable D2.7 and can be found there. However, support of GNS transactions implies an extension of the CALL_ID object defined in standard ASON/GMPLS signalling. Specific format of the new GNS CALL_EXT object is provided in section 4.3.

[2] In ASON a call is an association between two or more users and one or more domains that supports an instance of a service through one or more domains

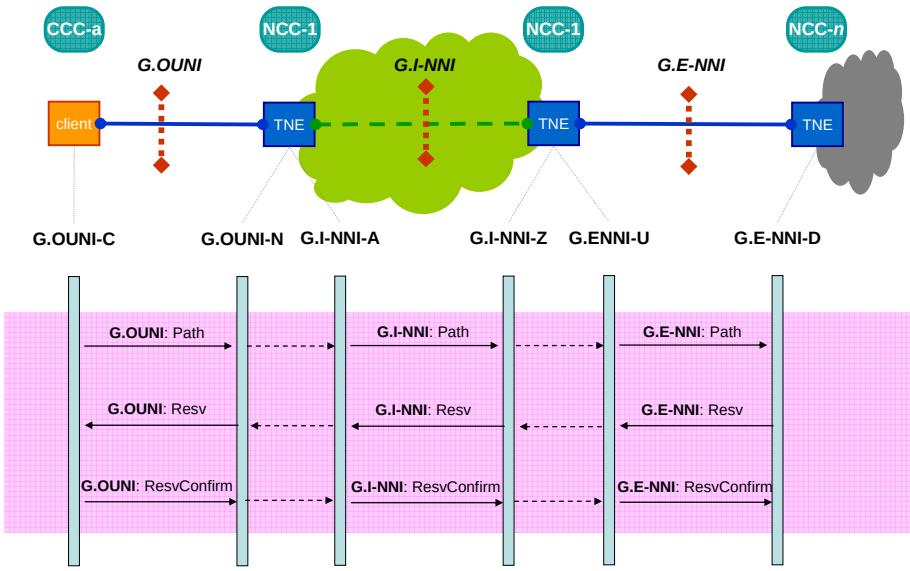


Figure 4-2: G^{2}RSVP-TE message mapping across the network reference points.

Due to the scope and processing rules of the CALL_ID object, the control of GNS transactions is narrowed to the network boundary nodes (i.e. those hosting the G.OUNI-N and the G.E-NNI-U and G.E-NNI-D), in which the G^{2}MPLS Network Call Control functionalities need to be implemented (ref. [PH-WP2-D2.1]).

4.2.2 Implicit Job destination (anycast)

Depending on the specific application, the network attachment point usually identified through the egress TNA and the egress label 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 be declared in the JSDL document (implicit declaration) and any Grid site matching the storage request could be a destination. Pure anycast in Phosphorus is limited to the G.OUNI interface, because on the ingress node of the G^{2}MPLS domain a path computation request occurs (local or remote to a computing server) with the purpose of defining the set of explicit routes that match the specified constraints and implement the GNS transaction segments, i.e. G^{2}MPLS calls. Possible change of the selected destination is possible in case of crankback procedures triggered by blocking conditions at the destination or along the selected path.

The implicit declaration of the destination of a job may be distinguished in two further sub-cases:

- participating Grid sites are specified, but the destination TNA is not declared. This implies that the G^{2}MPLS NCP will choose the best match in the set of the available network attachment points for the

selected site. The implicit destination turns into an explicit destination in case the set contains just one TNA.

- 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). This implies that the G^{2}MPLS NCP will pick at first the best site match in the set of the available Grid sites for that application and fulfilling the job requirements, and then the best match for the network attachment in the set of those available for the selected site.

The G^{2}MPLS signalling in case of implicit network destination implies that a new session must be set up from the ingress-node towards a new tunnel endpoint address when crankback and/or recovery occur in a domain. In fact, the destination address is the main piece of information in the LSP_TUNNEL SESSION object and cannot be modified, being it the root key for a (completed or on-going) reservation on the hosting nodes. The Being the ingress and/or border nodes the main coordinators of this special signalling procedure, they need to treat transparently grid resource description object in order to determine a new suitable destination endpoint. In this document a specific new object is introduced, i.e. the GNS_UNI object, to transport the end-to-end Grid resource requests. It will be transparent across the G.OUNI and the G.E-NNI, like in case of the GENERALIZED_UNI object in ASON DCM. On the contrary, it will be treated as opaque information at the G.I-NNI.

4.2.3 Advance reservations

Support of advance reservations in signalling may be split in two resource domains:

- head-end/far-end Grid resources
- network resources among them

Availability of resource calendars for both classes specified in section 3.2 can enable the enforcement of policies on advance reservation guarantee in signalling.

Origin of the time schedule for the reservations is the user or the middleware, which usually specify the start time of a certain service. In case this time schedule is missing an immediate reservation is assumed.

Duration of the service can be inferred by the time parameters of the job description. If not specified the resources are assumed to be reserved for long term call/connections until explicitly released. The trigger for release is always the middleware upon explicit command by the user or completion of a job in all its parts.

Time guarantees are just part of SLAs specified in WebService Agreements and could also be declared as extensions to JSDL documents. Therefore, they can be mapped in specific protocol objects related to the time values of a call. An example of wasg SLA for advance reservation compute job is provided in the following.

```
<wsag:GuaranteeTerm wsag:Name="ADVANCE_RESERVATION_GUARANTEE">
        <wsag:ServiceScope wsag:ServiceName="RESERVATION_SERVICE" />
        <wsag:ServiceLevelObjective>
                <wsag:KPITarget>
                        <wsag:KPIName>ADVANCE_RESERVATION</wsag:KPIName>
                        <wsag:CustomServiceLevel>
                                <ext:ReservationTime>
                                        2007-03-15T11:53:41.921+01:00
                                </ext:ReservationTime>
                        </wsag:CustomServiceLevel>
                </wsag:KPITarget>
        </wsag:ServiceLevelObjective>
        <wsag:BusinessValueList />
</wsag:GuaranteeTerm>
```

The reservation process is initiated and coordinated by the Grid scheduler through a sequence of phases, including:

- negotiation with the selected resources (local and remote Grid plus network in the middle) of a common timeframe for allocating the respective job parts;
- initial advance reservation of the agreed service segments, which implies a temporary booking of the service segment;
- final commit of the advance reservation, which results in the final booking of resources on the different service segments after all the initial booking have been positively acknowledged and correlated.

Then, upon the occurrence of the advance reservation time, resources must be available to the booked consumers.

The G^{2}MPLS signalling could be adapted to this negotiation phase by providing:

- an immediate processing and reservation of the advance reservation request for selected Grid and network resources, which results just in a booking of resources for later use. In terms of RSVP protocol this corresponds to the 3-tier session of Path – Resv - Resv Confirm.
- a later “service activation” tier (e.g. in the form of a Path Activate message with just the identifier of the session and the activation flag container) to be issued by the G.OUNI-C just before the execution of the job at the committed time. G.OUNI-C manager could asynchronously generate the Path activation event upon the expiration of a timer or it could receive am explicit command by the Grid middleware.

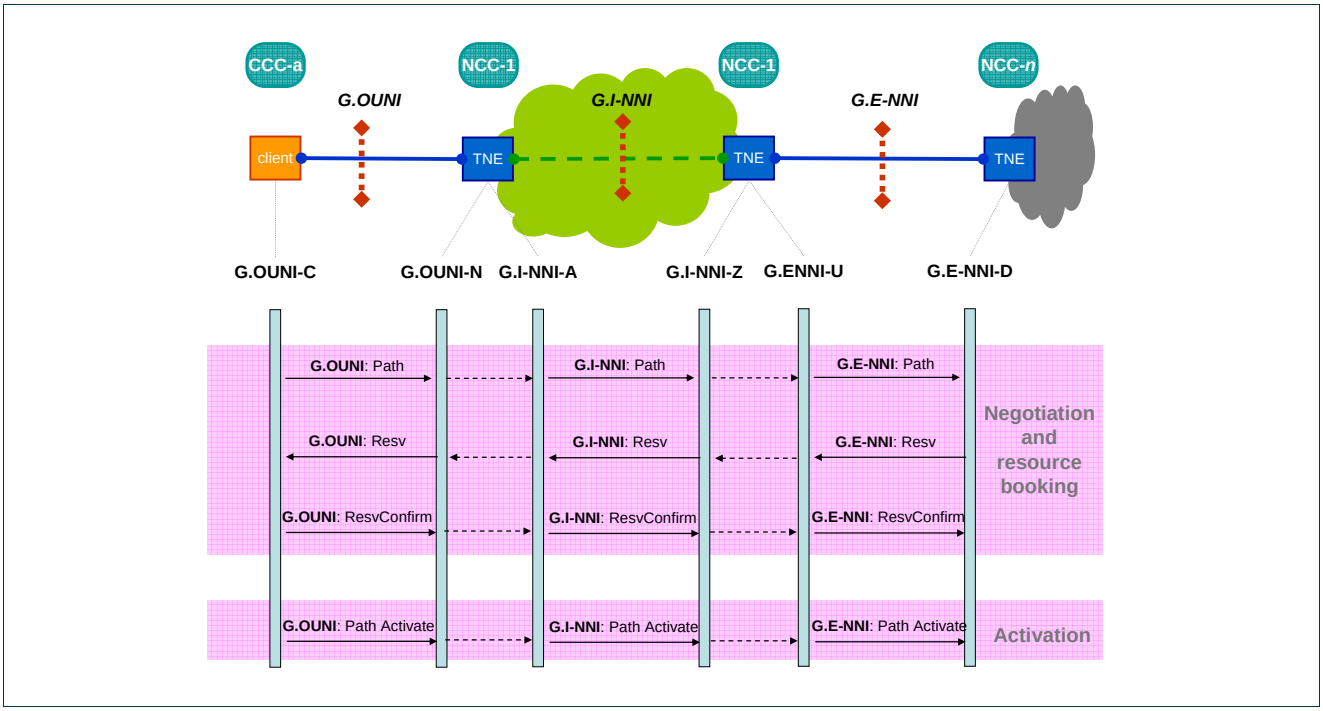


Figure 4-3: G^{2}RSVP-TE message sequence for advance reservation signalling and activation.

4.2.4 Full-optical TE signalling extensions in G^{2}MPLS

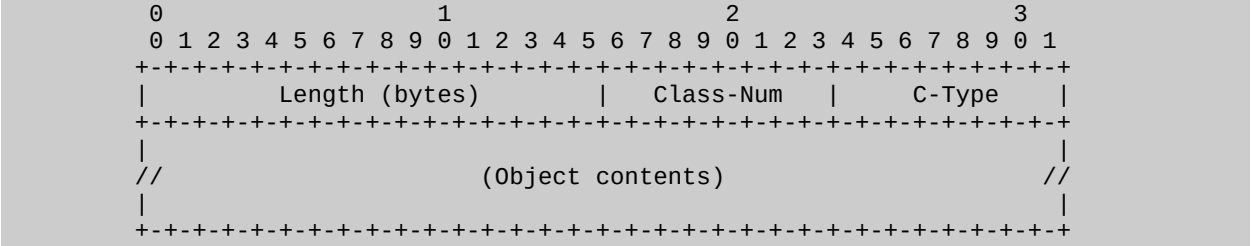
The architectural framework presented in [PH-WP2-D2.1] about optical impairments experimented by full-optical networks as a major impact on routing protocol and constrained based routing algorithms. Signalling occurs at a later stage, once an explicit route has been determined hopefully matching the fixed optimization objectives. In case of special requirements on wavelength selection in full optical networks, e.g. label continuity or limited wavelength conversion of the nodes, the explicit route could specify the source route up to the label detail. For this purpose routing protocol should aid the path computation algorithm by flooding wavelength allocation maps for each all-optical TE-link. The signalling protocol in this scenario has no special procedure, but it simply checks the status of the ERO label on the specified TE-link.

However, in case the routing information on wavelength is not published because of DCN traffic or LSDB size limitations, it is the signalling protocol the main responsible of matching of the fixed objectives. This case might also occur in case of recovery, above all if a segment recovery (span or local-to-egress) is activated. Many research efforts are ongoing in this field and most of them require limited protocol extensions. In fact a proper combination of the suggested label and the label set could provide solutions to the full-optical signalling problem. However, other proposals refer to the signalling of suggested vectors of candidate labels with a per label ranking. These solutions implement a kind of distributed shortest path algorithm according to the wavelength constraint, with a minimization objective established in the minimum number of wavelength conversions.

Due to the availability of the wavelength status bit-masks in G^{2}MPLS routing as specified in section 3.3, any signalling extension for full-optical networks is left for further studies and later consideration.

4.3 G^{2}MPLS extensions for RSVP-TE (G^2.RSVP-TE)

An RSVP message consists of a common header, followed by a body consisting of a variable number of variable-length, typed "objects". Every object consists of one or more 32-bit words with a one-word header, with the following format:



- Length (16 bits)
  - A field containing the total object length in bytes. Must always be a multiple of 4, and at least 4.
- Class-Num (8 bits)
  - Identifies the object class. Each object class has a name (ref. Appendix B) and different format types
- C-Type
  - Object type, unique within Class-Num.

The maximum object content length is 65528 bytes. The Class-Num and C-Type fields may be used together as a 16-bit number to define a unique type for each object.

The high-order two bits of the Class-Num is used to determine what action a node should take if it does not recognize the Class-Num of an object as specified in Table 4-9.

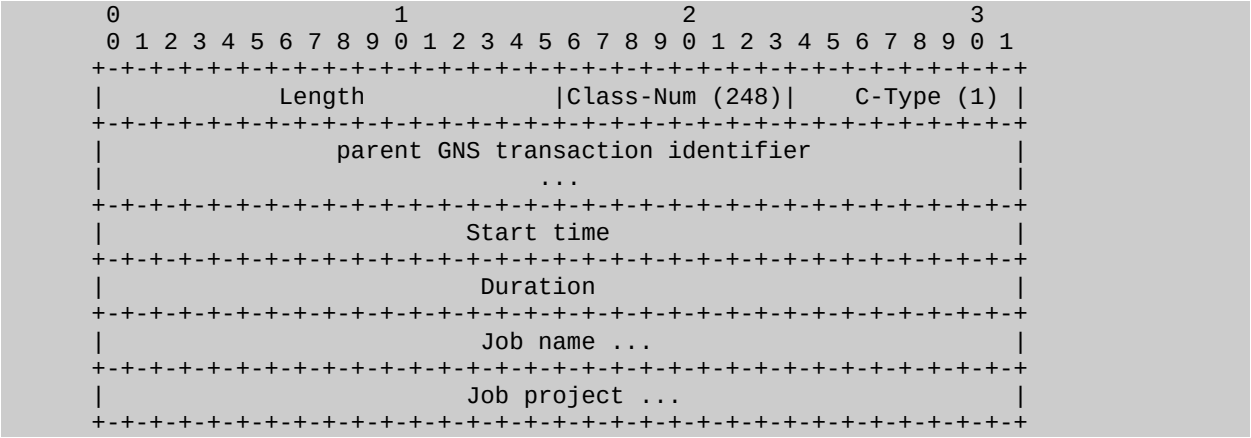
C-Num			Error when the object is not recognized
0bbbbbbb	0…119	Standards Action	The entire message should be rejected and an "Unknown Object Class" error returned.
	120…123	Expert Review	
	124…127	Vendor Private Use	
10bbbbbb	128…183	Standards Action	The node should ignore the object, neither forwarding it nor sending an error message.
	184…187	Expert Review	
	188…191	Vendor Private Use	
11bbbbbb	192…247	Standards Action	The node should ignore the object but forward it, unexamined and unmodified, in all messages resulting from this message.
	248…251	Expert Review	
	252…255	Vendor Private Use	

Table 4-9: IANA policy for C-Num allocation in RVSP.

G^{2}MPLS extensions for (G.OUNI, G.I-NNI and G.E-NNI) signalling results in brand new classes and addition of further types to standard RSVP objects just defined for MPLS, GMPLS; ASON DCM, OIF UNI and OIF E-NNI.

4.3.1 GNS_CALL_EXT Class (C-num = 248)

Just one object is defined (C-type = 1) in this class and it extends the call attributes specified through the standard CALL_ID object (C-num = 230), by adding a reference to the parent GNS transaction (job) ID, the time specification of the job and human-readable name for the job and the project.



The parent GNS transaction identifier (64 bits) is a unique identifier for the job originated by the Grid middleware system. It remains constant over the life of the job and call.

The start time is a 32-bit unsigned fixed-point number, in seconds relative to 0h on 1 January 1900. It is an NTP reduced timestamps (ref. IETF RFC4330). This field will overflow some time in 2036 (second 4,294,967,296). Similar actions as per NTP could be taken in that case if the proposed extension will be still in use. A null value of the start time must be assumed as an immediate reservation request.

The duration is a 32-bit unsigned fixed-point number specifying the job duration and consequently the call lifetime. Units are seconds. Duration must be considered a delta to add up to starting time. A null value must be assumed as an infinite (long-term) connection to be explicitly released upon command.

Job name is a null padded string derived from the JSDL JobIdentity.JobName element. It may not be unique to a particular session.

Job project is a null padded string derived from the JSDL JobIdentity.JobProject element. The project could be used by accounting systems or access control systems.

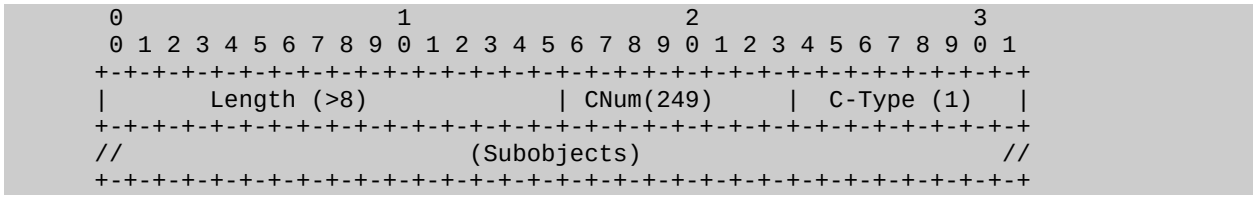
The GNS_CALL_EXT object appears in all RSVP messages in which a CALL_ID is present (Path, Resv, PathTear, PathErr, Notify). Its presence makes a standard call a G^{2}MPLS call and piggybacks the GNS transaction construct.

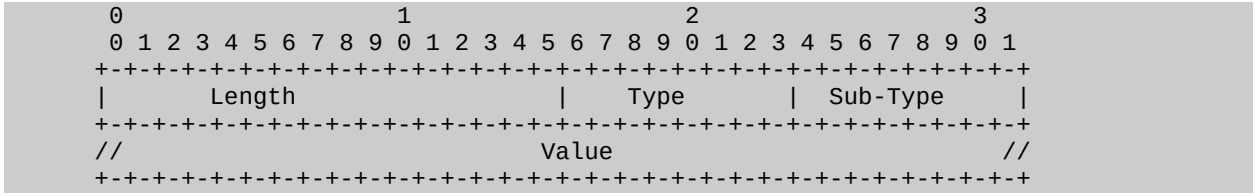
Error codes related to the processing of this object are the same as per call management (code 32, sub-codes 1-4, ref. Appendix B).

4.3.2 GNS_UNI Class (C-num = 249)

The GNS_UNI class is a generalized container of the Grid specific information derived from the JSDL document and transported end-to-end up to the G.OUNI interface. Just one object is defined for this class (C-Type = 1) and the different sub-objects are a series of variable-length data items.

The GNS_UNI object appears in the RSVP Path message like the GENERALIZED_UNI. No specific error code is identified for this object.

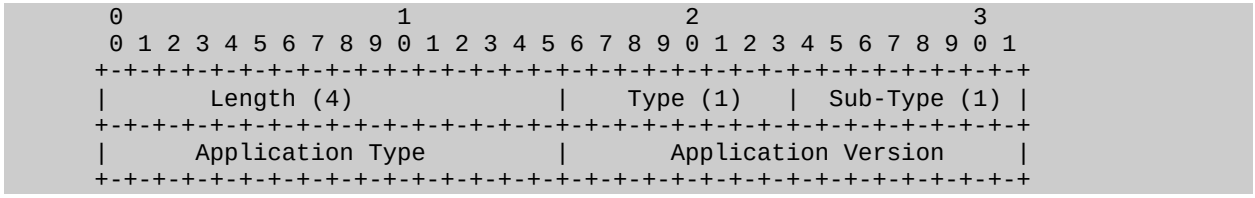




The following sub-objects are defined.

4.3.2.1 *Type 1: Application*

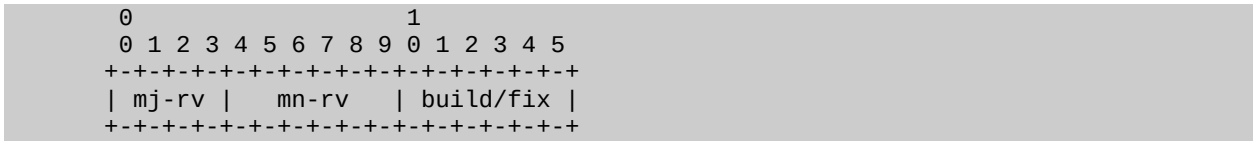
This object is optional and derives from the JSDL Application Element.



Application Type is the identifier of the software to be executed in the remote site and can assume one of the following values in the grid_software_t

- Wisdom 0x0001
- Kodavis 0x0002
- TOPS 0x0003
- DDSS 0x0004
- INCA 0x0005
- other 0xFFFF

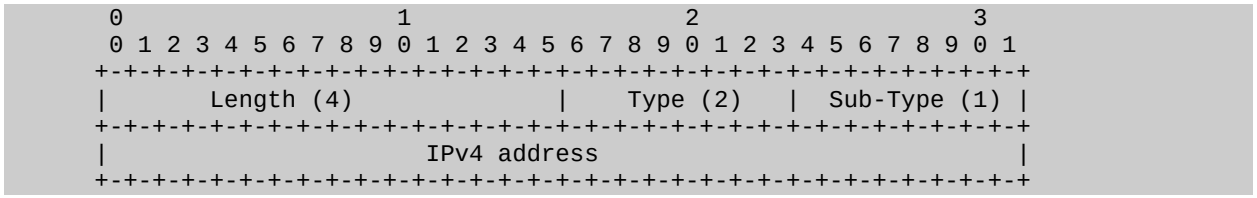
Application Version (16 bits) provides the specific version of the service in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)



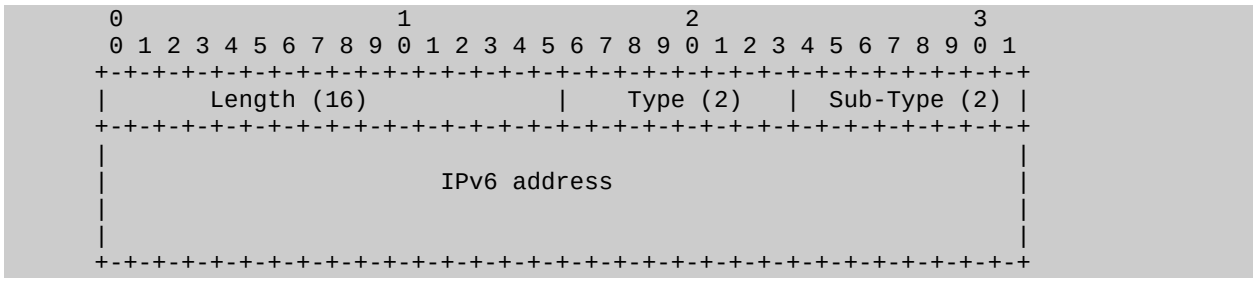
4.3.2.2 *Type 2: Candidate Hosts*

This object is optional and derives from the JSDL CandidateHosts in Resource Element. It is a list of addresses in the IPv4 (32bits), IPv6 (128 bits) and NSAP (160 bits) formats, defined through three different subtypes. The list composition is obtained by inserting more than one Grid Candidate Hosts TLV in the GNS_UNI object.

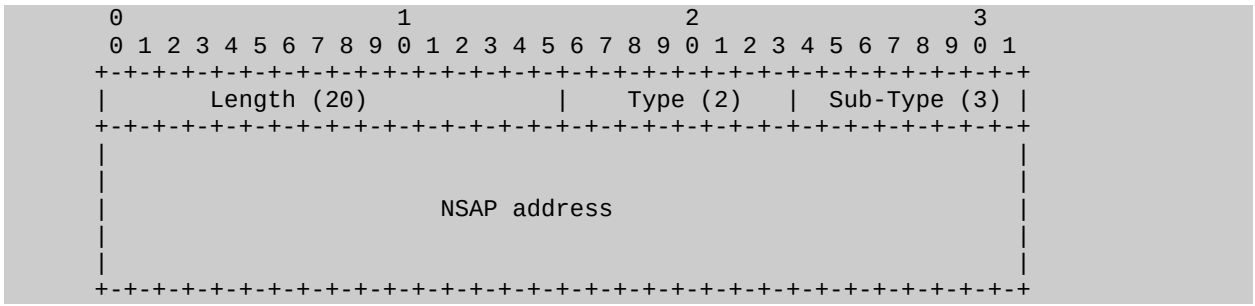
subtype 1: IPv4



subtype 2:IPv6

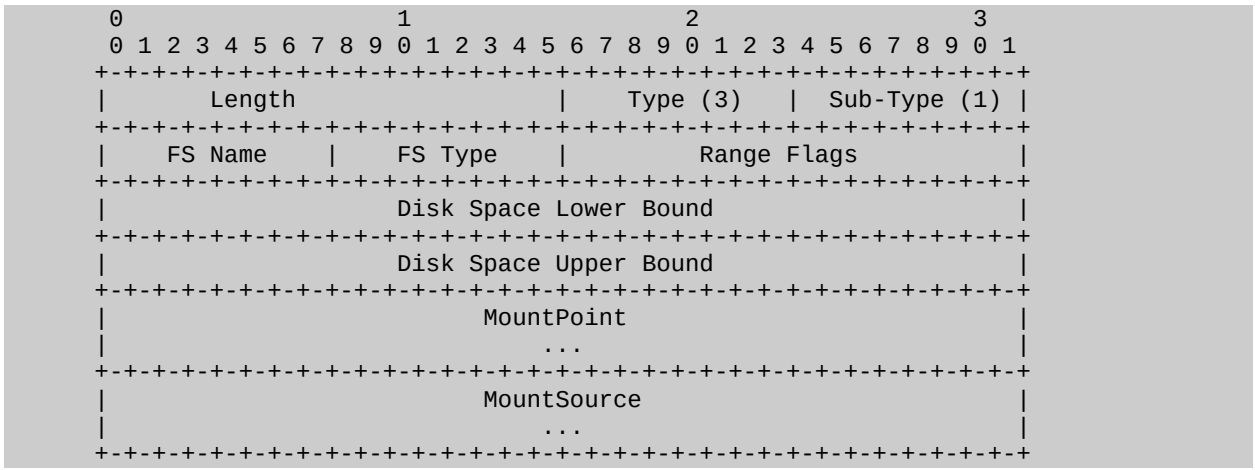


subtype 3:NSAP



4.3.2.3 *Type 3: File System Resources*

This object is optional and derives from the JSDL FileSystem in Resource Element.



FS Name is the symbolic name of the filesystem (8 bits) with values from the g2mpls_FSName_t:

- home 0x01
- root 0x02
- scratch 0x03
- tmp 0x04
- other 0xFF

FS Type is the common type of filesystem (8 bits) with values from the grid_fs_type_t (8 bits)

- swap 0x01
- temporary 0x02
- spool 0x03
- normal 0x04

- other 0xFF

Disk Space is specified with two 32 bit integers (units is byte) and two flags for the inclusion or not of the boundary values

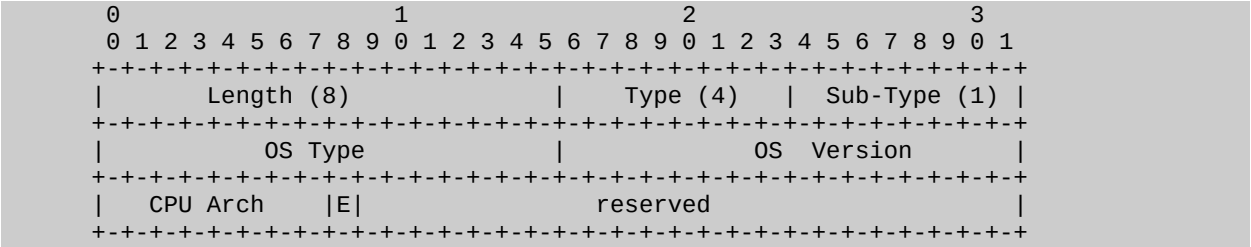
- Lower Bound represents the lower bound in the range of free disk space required by the job.
- Upper Bound represents the upper bound in the range of free disk space required by the job.
- Flags can assume the following values
  - Lower Bound included 0x01
  - Upper Bound included 0x10

Mount Point is a null padded string identifying the local mount point of the filesystem

Mount Source a null padded string identifying the remote (NFS) mount point of the filesystem

4.3.2.4 *Type 4: System Capabilities*

This object is optional and derives from the JSDL OperatingSystem and CPUArchitecture in Resource Element.



OS Type (16 bits) contains the name of the operating system and can assume one of the values in the grid_os_t defined in section 3.1.7;

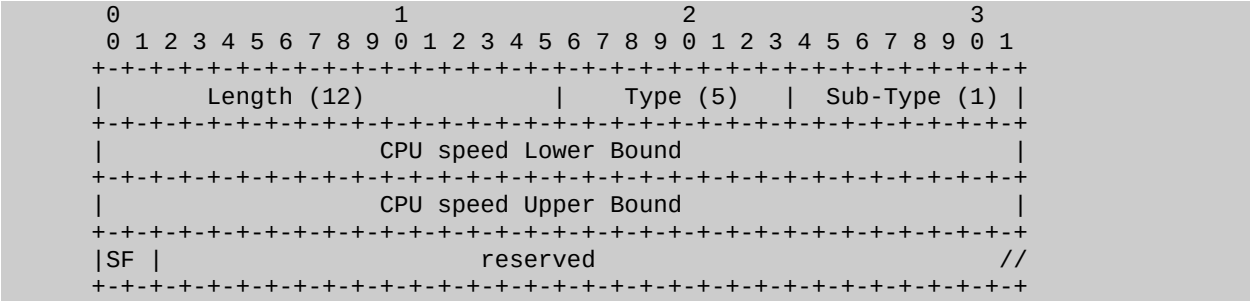
OS Version (16 bits) provides the specific version of the service in a generalized hierarchical 3 digits format: major version number (4 bits) – minor version number (6 bits) – build/fix version number (6 bits)

CPU Arch (8 bits) defines the CPU architecture required by the job in the execution environment and can assume values in grid_cpu_arch_t (8 bits) as defined in section 3.1.7. If this is not present then it is not defined and the consuming system may choose any value.

The bit E is a flag indicating whether the job must have exclusive access to the resources allocated to it by the consuming system.

4.3.2.5 *Type 5: Individual CPU Speed*

This object is optional and derives from the JSDL IndividualCPUSpeed in Resource Element.



IndividualCPUSpeed is specified with two 32 bit integers (units is hertz) and two flags for the inclusion or not of the boundary values

- Lower Bound represents the lower bound in the speed range for each CPU required by the job.
- Upper Bound represents the upper bound in the speed range for each CPU required by the job.
- Speed Flags (SF) can assume the following values
  - Lower Bound included 0x01
  - Upper Bound included 0x10

4.3.2.6 *Type 6: Individual CPU Time*

This object is optional and derives from the JSDL IndividualCPUTime in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.7 *Type 7: Individual CPU Count*

This object is optional and derives from the JSDL IndividualCPUCount in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.8 *Type 8: Individual Network Bandwidth*

This object is optional and derives from the JSDL IndividualNetworkBandwidth in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.9 *Type 9: Individual Physical Memory*

This object is optional and derives from the JSDL IndividualPhysicalMemory in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.10 *Type 10: Individual Virtual Memory*

This object is optional and derives from the JSDL IndividualVirtualMemory in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.11 *Type 11 Individual Disk Space*

This object is optional and derives from the JSDL IndividualDiskSpace in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.12 *Type 12: Total CPU Time*

This object is optional and derives from the JSDL TotalCPUTime in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.13 *Type 13: Total CPU Count*

This object is optional and derives from the JSDL TotalCPUCount in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.14 *Type 14: Total Physical Memory*

This object is optional and derives from the JSDL TotalPhysicalMemory in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.15 *Type 15: Type 7: Total Virtual Memory*

This object is optional and derives from the JSDL TotalVirtualMemory in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.16 *Type 16: Total Disk Space*

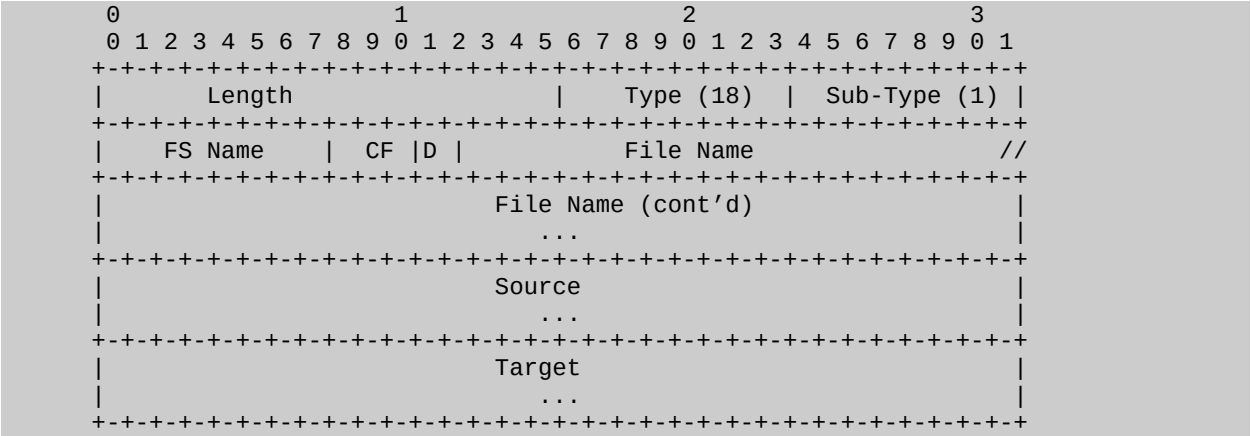
This object is optional and derives from the JSDL TotalDiskSpace in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.17 *Type 17: Total Resource Count*

This object is optional and derives from the JSDL TotalResourceCount in Resource Element. The format is similar to Individual CPU Speed for the Range definition.

4.3.2.18 *Type 18: Data Staging*

This object is optional and derives from the JSDL DataStaging Element.



FS Name is the symbolic name of the filesystem (8 bits) with values from the g2mpls_FSName_t:

- home 0x01
- root 0x02
- scratch 0x03
- tmp 0x04
- other 0xFF

CreationFlag (3 bits) is a field used to specify the action to be performed on the file staged in or out with values from the grid_ds_action_t:

- overwrite 0b001
- append 0b010

- dontOverwrite 0b100

Flag D (DeleteOnTermination) determines whether the file should be deleted after the job terminates. If true the file is deleted after the job terminates or after the file has been staged out. Default value depends on the consuming system but it can be assumed false.

FileName is a null padded string identifying a relative path in the execution host for a local file or directory to be used for staging.

Source is a null padded string (URI compliant with IETF RFC3986 specification) identifying the location (host) in which the file or directory to be staged in can be found. Stage in occurs before job starting. If this element is not present then the file does not have to be staged in.

Target is a null padded string (URI compliant with IETF RFC3986 specification) identifying the location (host) in which the file or directory must be staged out. Stage in occurs after job end. If this element is not present then the file does not have to be staged out.

4.3.3 Modified Message formats

The G.RSVP-TE messages are listed in this section with G^{2}MPLS additional objects highlighted in red boxes.

4.3.3.1 *Path*

```
<Path Message> ::=      <Common Header>
                        [ <INTEGRITY> ]
                        [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                        [ <MESSAGE_ID> ]
                        <SESSION>
                        <IF_ID_RSVP_HOP>
                        <TIME_VALUES>
                        [ <EXPLICIT_ROUTE> ]
                        <LABEL_REQUEST>
                        <CALL_OPS>
                        <CALL_ID>
                        <GNS_CALL_EXT>
                        [ <PROTECTION> ]
                        [ <LABEL_SET> ... ]
                        [ <SESSION_ATTRIBUTE> ]
                        [ <NOTIFY_REQUEST> ]
                        [ <ADMIN_STATUS> ]
                        <GENERALIZED_UNI>
                        <GNS_UNI>
                        [ <POLICY_DATA> ... ]
                        <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE>
                         <SENDER_TSPEC>
                         [ <ADSPEC> ]
                         [ <RECORD_ROUTE> ]
                         [ <SUGGESTED_LABEL> ]
                         [ <RECOVERY_LABEL> ]
```

The format of the sender description for bidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE>
                         <SENDER_TSPEC>
                         [ <ADSPEC> ]
                         [ <RECORD_ROUTE> ]
                         [ <SUGGESTED_LABEL> ]
                         [ <RECOVERY_LABEL> ]
                         <UPSTREAM_LABEL>
```

4.3.3.2 *Resv*

```
<Resv Message> ::=       <Common Header>
                         [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION>
                         <IF_ID_RSVP_HOP>
                         <TIME_VALUES>
                         <CALL_OPS>
                         <CALL_ID>
                         <GNS_CALL_EXT>
                         [ <RESV_CONFIRM> ]
                         [ <SCOPE> ]
                         [ <NOTIFY_REQUEST> ]
                         [ <ADMIN_STATUS> ]
                         [ <POLICY_DATA> ... ]
                         <STYLE>
                         <flow descriptor list>
```

```
    <flow descriptor list> ::= <FF flow descriptor list>
                             | <SE flow descriptor>
```

```
    <FF flow descriptor list> ::= <FLOWSPEC>
                                  <FILTER_SPEC>
                                  <LABEL>
                                  [ <RECORD_ROUTE> ]
                                | <FF flow descriptor list>
                                <FF flow descriptor>
```

```
<FF flow descriptor> ::= [ <FLOWSPEC> ]
                         <FILTER_SPEC>
                         <LABEL>
                         [ <RECORD_ROUTE> ]

<SE flow descriptor> ::= <FLOWSPEC>
                         <SE filter spec list>

<SE filter spec list> ::= <SE filter spec>
                         | <SE filter spec list> <SE filter spec>

<SE filter spec> ::=     <FILTER_SPEC>
                         <LABEL>
                         [ <RECORD_ROUTE> ]
```

4.3.3.3 *PathTear*

```
<PathTear Message> ::= <Common Header>
                       [ <INTEGRITY> ]
                       <SESSION>
                       <IF_ID_RSVP_HOP>
                       <CALL_OPS>
                       <CALL_ID>
                       <GNS_CALL_EXT>
                       [ <sender descriptor> ]

<sender descriptor> ::= (see earlier definition)
```

4.3.3.4 *ResvTear*

Not used.

4.3.3.5 *PathErr*

```
<PathErr message> ::= <Common Header>
                      [ <INTEGRITY> ]
                      <SESSION>
                      <ERROR_SPEC>
                      [ <POLICY_DATA> ...]
                      <CALL_OPS>
                      <CALL_ID>
                      <GNS_CALL_EXT>
                      [ <sender descriptor> ]
```

```
          <sender descriptor> ::= (see earlier definition)
```

4.3.3.6 *ResvErr*

Not used.

4.3.3.7 *ResvConf*

```
          <ResvConf message> ::= <Common Header>
                                 [ <INTEGRITY> ]
                                 <SESSION>
                                 <ERROR_SPEC>
                                 <RESV_CONFIRM>
                                 <STYLE>
                                 <flow descriptor list>


          <flow descriptor list> ::= (see earlier definition)
```

4.3.3.8 *Notify*

```
          <Notify message>  ::= <Common Header>
                                [<INTEGRITY>]
                                [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                                [ <MESSAGE_ID> ]
                                <ERROR_SPEC>
                                <notify session list>


       <notify session list> ::= [ <notify session list> ]
                                  <upstream notify session> |
                                  <downstream notify session>


   <upstream notify session> ::= <SESSION>
                                 <CALL_ID>
                                 <GNS_CALL_EXT>
                                 [ <ADMIN_STATUS> ]
                                 [<POLICY_DATA>...]
                                 <sender descriptor>


  <downstream notify session> ::= <SESSION>
                                  <CALL_ID>
                                  <GNS_CALL_EXT>
                                  [<POLICY_DATA>...]
                                  <flow descriptor list>
```

4.3.3.9 *Hello*

Unmodified.

4.3.3.10 *ACK/NACK*

Unmodified.

4.3.3.11 *Srefresh*

Unmodified.

5 Closing notes

This document specifies the Grid extensions for routing and signalling protocols. These extensions are needed to enhance a GMPLS Network Control Plane towards G^{2}MPLS.

Modelling of the Grid resources and job specifications was the preliminary activity, by which the main information elements have been identified. Then, objects mapping and adaptation/extension for the standard ones has been provided.

This deliverable is intended to be the main protocol reference for any developers of the G^{2}MPLS open source stack and complements the architectural contents provided by D2.1, D2.6 and (preliminarily) D2.7.

Improvements to the extensions set could derive in the future from the completion of the WP2 development activities and the execution of the test-bed validations. Further feedback channels could derive from the dissemination and standardization activities supported by the contents of this document. Any major issues identified during the project lifecycle as a follow up of the aforementioned activities will result in a revision of the deliverable and issue of an amendment.

6 References

6.1 Normative references

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

[ASON-DEF] ITU-T G.8081/Y.1353 Recommendations, “Definitions and Terminology for Automatically Switched Optical Networks (ASON)”, 2004.

[ASON-REQ] ITU-T G.807/Y.1302 Recommendations, “Requirements for the Automatic Switched Transport Network (ASTN)”, 2001.

[ASON-DCN] ITU-T G. 7712/Y.1703 Recommendations, “Data Communication Network”, 2003.

[ASON-DCM] ITU-T G.7713/Y.1704 Recommendations, “Generalised Distributed Connection Management”, 2001.

[ASON-PNNI] ITU-T G.7713.1/Y.1704 Recommendations, “Distributed Call and Connection Management – PNNI Implementation”, 2003.

[ASON-RSVP] ITU-T G.7713.2/Y.1704 Recommendations, “Distributed Call and Connection Management – GMPLS RSVP-TE Implementation”, 2003.

[ASON-LDP] ITU-T G.7713.3/Y.1704 Recommendations, “Distributed Call and Connection Management – GMPLS CR-LDP Implementation”, 2003.

[ASON-AD] ITU-T G.7714/Y.1705 Recommendations, “Generalised automatic discovery techniques”, 2001

[ASON-AD-SDH] ITU-T G.7714.1/Y.1705.1 Recommendations, “Protocol for automatic discovery in SDH and OTN networks”, 2003.

[ASON-ROUT] ITU-T G.7715/Y.1706 Recommendations, “Architecture and requirements for routing in automatically switched optical networks”, 2002.

[ITU-G.841] ITU-T G.841, "Types and Characteristics of SDH Network Protection Architectures," 1998.

[ITU-G.808.1] ITU-T G.808.1, "Generic Protection Switching – Linear trail and subnetwork protection," 2003.

[IETF-RFC3945] E. Mannie (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, IETF RFC 3945, October 2004.

[IETF-RFC3471] L. Berger (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description”, IETF RFC 3471, January 2003.

[IETF-RFC3472] P. Ashwood-Smith (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions”, IETF RFC 3472, January 2003.

[IETF-RFC3473] L. Berger (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions”, IETF RFC 3473, January 2003.

[IETF-RFC4257] G. Bernstein (Ed.), “Framework for Generalized Multi-Protocol Label Switching (GMPLS)-based Control of Synchronous Digital Hierarchy/Synchronous Optical Networking (SDH/SONET) Networks”, IETF RFC 4257, December 2005.

[IETF-RFC4606] E. Mannie (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control“, IETF RFC 4606, August 2006.

[IETF-RFC4328] D. Papadimitriou (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Extensions for G.709 Optical Transport Networks Control”, IETF RFC 4328, January 2006.

[IETF-RFC4139] D. Papadimitriou (Ed.), “Requirements for Generalized MPLS (GMPLS) Signaling Usage and Extensions for Automatically Switched Optical Network (ASON)”, IETF RFC 4139, July 2005.

[IETF-RFC4208] G. Swallow (Ed.), “Generalize Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model”, IETF RFC 4208, October 2005.

[IETF-RFC4201] K. Kompella, Y. Rekhter, L. Berger, “Link Bundling in MPLS Traffic Engineering (TE)”, IETF RFC 4201, October 2005.

[IETF-RFC4202] K. Kompella (Ed.), “Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)”, IETF RFC 4202, October 2005.

[IETF-RFC3630] K. Katz (Ed.), “Traffic Engineering (TE) Extensions to OSPF Version 2”, IETF RFC 3630, September 2003.

[IETF-RFC4203] K. Kompella (Ed.), “OSPF Extensions in Support of Generalized Multi-Protocol Label Switching”, IETF RFC 4203, October 2005.

[IETF-RFC4652] D. Papadimitriou (Ed.), “Evaluation of Existing Routing Protocols against Automatic Switched Optical Network (ASON) Routing Requirements”, IETF RFC 4652, October 2006.

[IETF-RFC4258] D. Brungard (Ed.), “Requirements for Generalized Multi-Protocol Label Switching (GMPLS) Routing for the Automatically Switched Optical Network (ASON)”, IETF RFC 4258, November 2005.

[IETF-RFC4204] J. Lang (Ed.), “Link Management Protocol (LMP)”, IETF RFC 4204, October 2005.

[IETF-RFC4207] J. Lang (Ed.), “Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) Encoding for Link Management Protocol (LMP) Test Messages”, IETF RFC 4207, October 2005.

[IETF-RFC4209] A. Fredette (Ed.), “Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems”, IETF RFC 4209, October 2005.

[IETF-RFC4426] J. Lang (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification", IETF RFC 4426, March 2006.

[IETF-RFC4427] E. Mannie (Ed.), “Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)”, IETF RFC 4427, March 2006.

[IETF-RFC4428] D. Papadimitriou (Ed.), “Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)”, IETF RFC 4428, March 2006.

[IETF-SEGREC] L. Berger, I. Bryskin, D. Papadimitriou, A. Farrel, "GMPLS Based Segment Recovery", IETF Draft, work in progress, draft-ietf-ccamp-gmpls-segment-recovery-03.txt, October 2006.

[IETF-IDREC] 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.

[IETF-E2ESIGREC] J.P. Lang (Ed.), "RSVP-TE Extensions in support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", IETF Draft, work in progress, draft-ietf-ccamp-gmpls-recovery-e2e-signaling-04.txt, October 2006.

[IETF-RFC4726] A. Farrel (Ed.), "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", IETF RFC 4726, November 2006.

[IETF-RFC4783] L. Berger (Ed.), "GMPLS - Communication of Alarm Information", IETF RFC 4783, December 2006.

[IETF-RFC3477] K. Kompella, Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol – Traffic Engineering (RSVP-TE)", IETF RFC 3477, January 2003.

[IETF-RFC3480] K. Kompella, Y. Rekhter, A. Kullberg, "Signalling Unnumbered Links in CR-LDP (Constraint-Routing Label Distribution Protocol)", IETF RFC 3480, February 2003.

[IETF-RFC4655] A. Farrel J.-P. Vasseur, J.Ash, "A Path Computation Element (PCE) – Based Architecture", IETF RFC 4655, August 2006.

[IETF-RFC4940] K. Kompella, B. Fenner, "IANA Considerations for OSPF" IETF RFC 4940, June 2007.

[IETF-RFC3936] K. Kompella, J. Lang, "Procedures for Modifying the Resource reSerVation Protocol (RSVP)", IETF RFC 3936, October 2004.

[IETF-RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", IETF RFC 3986, January 2005.

[OIF-UNI1.0] Optical Interworking Forum, "User Network Interface (UNI) 1.0 Signalling Specification", http://www.oiforum.com/public/documents/OIF-UNI-01.0.pdf.

[OIF-UNI1.0R2-COMM] Optical Interworking Forum, "UNI 1.0 Signaling Specification, Release 2: Common Part", http://www.oiforum.com/public/documents/OIF-UNI-01.0-R2-Common.pdf.

[OIF-UNI1.0R2-RSVP] Optical Interworking Forum, "RSVP Extensions for User Network Interface (UNI) 1.0 Signaling, Release 2", http://www.oiforum.com/public/documents/OIF-UNI-01.0-R2-RSVP.pdf.

[OIF-UNI2.0-COMM] Optical Interworking Forum, "UNI 2.0 Signaling Specification: Common Part", draft OIF2005.204.02, work in progress.

[OIF-UNI2.0-RSVP] Optical Interworking Forum, "RSVP Extensions for User Network Interface (UNI) 2.0 Signaling", draft OIF2005.205.01, work in progress.

[OIF-E-NNI-Sig-1.0] Optical Interworking Forum, "Intra-Carrier E-NNI 1.0 Signaling Specification", http://www.oiforum.com/public/documents/OIF-E-NNI-Sig-01.0-rev1.pdf.

[OIF-E-NNI-Sig-2.0] Optical Interworking Forum, "Intra-Carrier E-NNI 2.0 Signaling Specification", draft OIF2005.381.03, work in progress.

[OIF-E-NNI-Rtr-1.0] Optical Interworking Forum, "External Network-Network Interface (E-NNI) OSPF-based Routing - 1.0", http://www.oiforum.com/public/documents/OIF-ENNI-OSPF-01.0.pdf.

[OGF-GFD80] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich, "The Open Grid Services Architecture, Version 1.5", GFD-I.080 http://forge.gridforum.org/projects/ogsa-wg, July 2006.

[W3C-WDSL] Web Services Description Language (WSDL) 1.1, W3C, http://www.w3.org/TR/wsdl.

[UDDI] UDDI: Universal Description, Discover and Integration of Business for the Web, http://www.uddi.org.

[WSIL] "Web Services Inspection Language (WSIL), http://xml.coverpages.org/IBM-WS-Inspection-Overview.pdf.

[WSFL] Web Services Flow Language (WSFL)", http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf .

[GLUE] S. Andreozzi (Ed.), "GLUE Schema Specification – version 1.3", Draft 3 – 16 Jan 2007, http://glueschema.forge.cnaf.infn.it/Spec/V13.

[JSDL] A. Savva (Ed.), "Job Submission Description Language (JSDL) Specification, Version 1.0", GFD.056, http://forge.gridforum.org/projects/jsdl-wg, November 2005.

[JSDL-1.0-r14] A. Savva (Ed.), "Job Submission Description Language (JSDL) Specification, Version 1.0 (Errata draft 014)", GFD-R–P.056 (draft-ogf-jsdl-spec-1.0-errata-014), http://forge.gridforum.org/projects/jsdl-wg, October 2007.

6.2 Informational references

[PH-WP2-D2.1] Phosphorus deliverable D2.1, "The Grid-GMPLS Control Plane architecture".

[OGF-GNS] G. Clapp, T. Ferrari, D.B. Hoang, A. Jukan, M.J. Leese, P. Mealor, F. Travostino, "Grid Network Services", draft-ggf-ghpn-netserv-2, Grid High Performance Networking Research Group working draft, May 2005.

[OGF-GFD81] J. Treadwell (Ed.), "Open Grid Services Architecture Glossary of Terms Version 1.5", GFD-I.081, http://forge.gridforum.org/projects/ogsa-wg, July 2006.

[IETF-RFC4397] I. Bryskin (Ed.), "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within The Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", IETF RFC 4397, February 2006.

[VIOLA UBonn] Christoph Barz and Markus Pilz, "Co-Allocation of Compute and Network Resources in the VIOLA-Testbed", TERENA 2006

[IETF-RFC3444] A. Pras, J. Schoenwaelder, "On the Difference between Information Models and Data Models", IETF RFC 3444, January 2003

[GRID-PHYSIOLOGY] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Globus Project, 2002.

[PHOSPHORUS-D1.1] PHOSPHORUS Deliverable D.1.1, "Requirements and specifications of interfaces and architecture for interoperability between NRPS, GMPLS, Middleware".

[GN2-IDM] GÉANT2 Deliverable DJ.3.3.1, "GÉANT2 Bandwidth on Demand Framework and General Architecture".

[STRAND-01] J. Strand, A. Chiu, R. Tkach, "Issues For Routing In The Optical Layer", IEEE Communications Magazine, February 2001, pp.81-87.

[OGF-GDF56] D. Simeonidou, R. Nejabati (Editors), "Optical Network Infrastructure for Grid", GFD-I-036, http://forge.gridforum.org/projects/ghpn-rg/, August 2004.

[Agrawal] G.Agrawal, "Nonlinear Fiber Optics, Academic Press ,1995

7 Acronyms

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

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

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

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

Appendix A Summary of standard GMPLS and OIF TE routing extensions

This section summarizes the standard GMPLS and OIF TE routing extensions supported by G^{2}MPLS and implemented according to the respective standards.

References for this section are:

- base OSPF: IETF RFC2328,RFC2370
- OSPF-TE for MPLS: IETF RFC3630,
- OSPF-TE for GMPLS: IETF RFC4202, RFC4203
- OSPF-TE for ASON: IETF RFC4258, ITU-T G.7715, ITU-T G.7715.1
- OSPF-TE for OIF E-NNI: OIF ENNI-OSPF-01.0

This appendix reports state-of-the-art information and is provided just for quick reference by any developers of the G^{2}MPLS open source stack.

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS generic	Router Address	specifies a stable IP address of the advertising router that is always reachable if there is any connectivity to it; this is typically implemented as a "loopback address". this is known as the "router ID"	RFC3630	IP address	Optional	Exactly once	NetAdmin
GMPLS generic	Link type	defines the type of the link	RFC3630	1 - Point-to-point, 2 - Multi-access	Mandatory	Exactly once	NetAdmin
GMPLS generic	Link ID	identifies the other end of the link. For p2p links, this is the Router ID of the neighbour. For multi-access links, this is the interface address of the designated router. The Link ID is identical to the contents of the Link ID field in the Router LSA for these link types.	RFC3630	IP address	Mandatory	Exactly once	NCP
GMPLS generic	Local interface IP address	specifies the IP address(es) of the interface corresponding to this link. If there are multiple local addresses on the link, they are all listed in this sub-TLV	RFC3630	IP address	Optional	Exactly once	NetAdmin
GMPLS generic	Remote interface IP address	specifies the IP address(es) of the neighbour's interface corresponding to this link. If the Link Type of the link is Multi-access, the Remote Interface IP Address is set to 0.0.0.0; alternatively, an implementation MAY choose not to send this sub-TLV.	RFC3630	IP address	Optional	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS generic	Traffic engineering metric	specifies the link metric for traffic engineering purposes. This metric may be different than the standard OSPF link metric.	RFC3630	32-bit value	Optional	Exactly once	NetAdmin
GMPLS generic	Maximum bandwidth	specifies the maximum bandwidth that can be used on this link, in this direction(true link capacity, units are bytes per second)	RFC3630	32-bit IEEE FP	Optional	Exactly once	Eqpt
GMPLS generic	Maximum reservable bandwidth	specifies the maximum bandwidth that may be reserved on this link, in this direction (units are bytes per second). This may be greater than the maximum bandwidth (oversubscription).	RFC3630	32-bit IEEE FP	Optional	Exactly once	NetAdmin
GMPLS generic	Unreserved bandwidth	specifies the amount of bandwidth not yet reserved at each of the eight priority levels (bw[i] <= Maximum Reservable Bandwidth, units are bytes per second).	RFC3630	8 * 32-bit IEEE FP	Optional	Exactly once	Eqpt
GMPLS generic	Administrative group	bit mask (Resource Class/Color) in which each set bit corresponds to one administrative group assigned to the interface. A link may belong to multiple groups.	RFC3630	32-bit value	Optional	Exactly once	NetAdmin
GMPLS generic	Link Local/Remote Identifiers	specifies TE link identifiers (local and remote) in case of unnumbered links. If the Remote Identifier is unknown, it is set to 0.	RFC4203	2 * 32-bit value	Optional	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS generic	Link Protection Type	represents the protection capability that exists for a link	RFC4203	0x01 Extra Traffic, 0x02 Unprotected, 0x04 Shared, 0x08 Dedicated 1:1, 0x10 Dedicated 1+1, 0x20 Enhanced, 0x40 Reserved, 0x80 Reserved	Optional	Exactly once	NetAdmin
GMPLS generic	Interface Switching Capability Descr. - Switching Capability	describes the link switching capability	RFC3471	1 (PSC-1), 2 (PSC-2), 3 (PSC-3), 4 (PSC-4), 51 (L2SC), 100 (TDM), 150 (LSC), 200 (FSC)	Optional	More than once	Eqpt

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS generic	Interface Switching Capability Descr. - Encoding Type	specifies the link encoding type	RFC3471, RFC4328	1 Packet, 2 Ethernet, 3 ANSI/ETSI PDH, 4 Reserved, 5 SDH ITU-T G.707 / SONET ANSI T1.105, 6 Reserved, 7 Digital Wrapper (non-standard path layer), 8 Lambda (photonic), 9 Fiber, 10 Reserved, 11 FiberChannel, 12 G.709 ODUk (Digital Path), 13 G.709 Optical Channel	Optional	More than once	Eqpt
GMPLS generic	Interface Switching Capability Descr. - Max LSP Bandwidth [prio]	specifies the smaller of the unreserved bandwidth at priority p	RFC4203	8 * 32-bit IEEE FP	Optional	More than once	NetAdmin
GMPLS generic	Interface Switching Capability Descr. - Min LSP Bandwidth	specifies the minimum bandwidth an LSP could reserve -- PSC & TDM ONLY	RFC4203	32-bit IEEE FP	Optional	More than once	NetAdmin
GMPLS generic	Interface Switching Capability Descr. - Interface MTU	specifies the maximum size of a packet that can be transmitted on this interface without being fragmented -- PSC ONLY	RFC4203	32-bit value	Optional	More than once	Eqpt
GMPLS generic	Interface Switching Capability Descr. - Indication	specifes whether the interface supports Standard or Arbitrary SDH -- TDM ONLY	RFC4203	32-bit value	Optional	More than once	Eqpt
GMPLS generic	Shared Risk Link Group	unordered list of 32 bits numbers that are the SRLGs that the link belongs to.	RFC4203	N * 32-bit values	Optional	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS ASON routing	Node IPv4 local prefix	advertises blocks of reachable address prefixes (IP+netmask) of a router as new sub-TLVs of the Node Attribute top level TLV. Each <Network mask, IPv4 Address> pair represents a reachable destination prefix hosted by the advertising Router ID. The local addresses that can be learned from TE LSAs i.e. router address and TE interface addresses SHOULD not be advertised in the node IPv4 local prefix sub-TLV.	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	N * {netmask 32-bit, IPv4 address 32-bit}	Optional	Exactly once	NCP
GMPLS ASON routing	Node IPv6 local prefix	advertises blocks of reachable address prefixes (IP+netmask) of a router as new sub-TLVs of the Node Attribute top level TLV. Each <Network mask, IPv6 Address> pair represents a reachable destination prefix hosted by the advertising Router ID. The local addresses that can be learned from TE LSAs i.e. router address and TE interface addresses SHOULD not be advertised in the node IPv4 local prefix sub-TLV.	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	N * {PrefixLength + PrefixOptions + zero padding + IPv6 Address Prefix}	Optional	Exactly once	NCP

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS ASON routing	Technology Specific Bandwidth Accounting	accounting on per timeslot basis using 32-bit tuples of the form <signal_type (8 bits); number of unallocated timeslots (24 bits)> may optionally be incorporated in the technology specific field of the ISCD TE link attribute when the switching capability field is set to TDM value.	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	N * {signal_type (8 bits) + number of unallocated timeslots (24 bits)}	Optional	Exactly once	Eqpt
GMPLS ASON routing	Local and Remote TE Router ID sub-TLV	The value field of this sub-TLV contains four octets of Local TE Router Identifier followed by four octets of Remote TE Router Identifier. The value of the Local and the Remote TE Router identifier SHOULD NOT be set to 0. This sub-TLV is optional and SHOULD only be included as part of the top level Link TLV if the Router_ID is advertising on behalf of more than one TE_Router_ID.	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	Local TE Router Identifier (32-bit) Remote TE Router Identifier (32-bit)	Optional	Exactly once	NetAdmin
GMPLS ASON routing	Local TE Router ID sub-TLV	a new sub-TLV of the (OSPFv2 TE LSA) top level Node Attribute TLV is introduced. This TLV associates the local prefixes (sub-TLV 3 and 4, see above) to a given TE Router_ID	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	Local TE Router Identifier (32-bit)	Optional	Exactly once	NetAdmin
GMPLS ASON routing	OSPF Downstream Associated Area ID	specific TLV indexing the (lower) area ID to which the RC's are capable to disseminate routing information is needed. OSPF Downstream Associated Area ID TLV is carried in the OSPF router information LSA [OSPF-CAP]	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	N * {Associated Area ID 32-bit}	Optional	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS ASON routing	Associated Area ID	Area ID associated to the incoming routing information. This additional information MAY be carried in opaque LSAs including the Router Address TLV, in opaque LSAs including the Link TLV, and in opaque LSAs including the Node Attribute TLV. It is used to solve the loop prevention problem.	draft-ietf-ccamp-gmpls-ason-routing-ospf-02.txt	Associated Area ID 32-bit	Optional	Exactly once	NetAdmin
OIF E-NNI routing	Local Node ID	Local Node ID sub-TLV (Type 32773) is included in an inter-domain or intra-domain TE link LSA to indicate the local end point of a link.	OIF-ENNI-OSPF-01.0	IPv4 address (32 bits)	Mandatory	Exactly once	NetAdmin
OIF E-NNI routing	Remote Node ID	Remote Node ID sub-TLV (Type 32774) is included in an inter-domain or intra-domain TE link LSA to indicate the remote end point of a link.	OIF-ENNI-OSPF-01.0	IPv4 address (32 bits)	Mandatory	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
OIF E-NNI routing	Sonet/SDH Interface Switching Capability Descriptor sub-TLV	Inherited from [RFC 4203], the Switching Capability field and the Encoding field MUST take the following values for Sonet/SDH interfaces: - Switching Capability (8 bits): value 100 (TDM). - Encoding (8 bits): value 5 for Sonet/SDH. - Reserved (16 bits): set to zero when sent and ignored when received. - Signal Type (8 bits): inherited from [RFC 3946], - Number of Unallocated Timeslots (24 bits):	OIF-ENNI-OSPF-01.0	signal type: 1 VT1.5 SPE / VC-11 2 VT2 SPE / VC-12 3 VT3 SPE 4 VT6 SPE / VC-2 5 STS-1 SPE / VC-3 6 STS-3c SPE / VC-4 21 STS-12c SPE/VC-4-4c 22 STS-48c SPE/VC-4-16c 23 STS-192c SPE/VC-4-64c	Optional	More than once	Eqpt
OIF E-NNI routing	TNA Address Sub-TLV	specifies one TNA address. Three possible formats are defined for the TNA address: IPv4, IPv6, or NSAP.	OIF-ENNI-OSPF-01.0	Addr length (8 bits) used as prefix mask TNA (32 bits, or 128 bits or 160 bit)	Optional	More than once	NetAdmin
OIF E-NNI routing	Node_ID	contains the node hosting this TNA address(es).	OIF-ENNI-OSPF-01.0	IPv4 address (32 bits)	Optional	More than once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
OIF E-NNI routing	General Capabilities	specifies the general capabilities of the RCD as advertised by the associated RC	OIF2005.313.01	Flag S (bit 0 and 1): - 0x00 – reserved - 0x01 – SONET switching-capable - 0x10 – SDH switching-capable - 0x11 – SONET and SDH switching-capable • Flag T (bit 2): - 0 – non-transit control domain - 1 – transit control domain • Flag M (bit 3): - 0 – not support branching for point-to-multipoint connections - 1 – support branching for point-to-multipoint connections	Optional	Exactly once	NetAdmin
OIF E-NNI routing	Hierarchy List	includes a list of Routing Controller IDs where each RC ID specifies a RC at a given hierarchical level. The list is arranged in the descending order of the hierarchy	OIF2005.313.01	{RC ID of the highest hierarchy(32-bit), …, RC ID of the lowest (this) hierarchy}	Optional	Exactly once	NetAdmin
OIF E-NNI routing	Ancestor RC ID	in a TE link to specify the local RC ID of an ancestor that is the local endpoint of a TE link at a higher level hierarchy. The Ancestor RC is used to identify the level of hierarchy associated with a border link.	OIF2005.313.01	IPv4 address (32 bits)	Optional	Exactly once	NetAdmin

Reference	Parameter	Description	Origin	Values	TE LSA Status	TE LSA	Owner
GMPLS all-optical extensions	BER estimate	specifies the exponent from the BER representation	RFC4209	32-bit value	Optional	Exactly once	NetAdmin
GMPLS all-optical extensions	Span length	represents the total length of the WDM span in meters	RFC4209	32-bit value	Optional	Exactly once	NetAdmin
GMPLS all-optical extensions	OSNR	specifies the value in dB of the signal to noise ratio	STRAND, D2.1	32-bit value	Optional	Exactly once	NetAdmin
GMPLS all-optical extensions	DPMD	is the fiber PMD parameter in ps per sqrt(km) of the k-th span in the circuit	STRAND, D2.1	32-bit IEEE FP	Optional	Exactly once	NetAdmin
GMPLS all-optical extensions	Amplifiers list	List of amplifiers traversed in the span, including their gain G and noise figure nSP	STRAND, D2.1	N * {gain (32 bits value in dB) + nSP (32 bits IEEE FP)}	Optional	Exactly once	NetAdmin
GMPLS all-optical extensions	Available wavelength mask	bitmask of the available wavelengths	STRAND, D2.1	N * {32 bits value}	Optional	Exactly once	NetAdmin

Appendix B Summary of RSVP protocol, base objects and GMPLS extensions

This section reports the base messages and objects of the RSVP protocol and extensions for GMPLS and ASON architectures.

References for this section are:

- base RSVP: IETF RFC2205,RFC2210,RFC2215,RFC2747,RFC2750,RFC2961
- RSVP for MPLS: IETF RFC3209,
- RSVP for GMPLS: IETF RFC3473, RFC3477
- RSVP for ASON: IETF RFC3474,
- RSVP for OIFUNI: RFC3476

This appendix reports state-of-the-art information and is provided just for quick reference by any developers of the G^{2}MPLS open source stack.

B.1 RSVP Common Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Vers  | Flags |    Msg Type   |         RSVP Checksum         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Send_TTL    |   (Reserved)  |         RSVP Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields in the common header are as follows:

- Vers: 4 bits
  - Protocol version number. This is version 1.
- Flags: 4 bits

- 0x01: Refresh (overhead) reduction capable
- 0x01-0x08: Reserved

- Msg Type: 8 bits
  - 1 = Path [RFC2205]
  - 2 = Resv [RFC2205]
  - 3 = PathErr [RFC2205]
  - 4 = ResvErr [RFC2205]
  - 5 = PathTear [RFC2205]
  - 6 = ResvTear [RFC2205]
  - 7 = ResvConf [RFC2205]
  - 13 = ACK [RFC2961]
  - 15 = Srefresh [RFC2961]
  - 20 = Hello [RFC3209]
  - 21 = Notify Message [RFC3473]
  - 30 = RecoveryPath [RFC-ietf-ccamp-rsvp-restart-ext-09.txt]
- RSVP Checksum: 16 bits
  - The one's complement of the one's complement sum of the message, with the checksum field replaced by zero for the purpose of computing the checksum. An all-zero value means that no checksum was transmitted.
- Send_TTL: 8 bits
  - The IP TTL value with which the message was sent.
- RSVP Length: 16 bits
  - The total length of this RSVP message in bytes, including the common header and the variable-length objects that follow.

B.2 RSVP Message formats

B.2.1 Path

```
<Path Message> ::=       <Common Header>
                         [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION>
                         <IF_ID_RSVP_HOP>
                         <TIME_VALUES>
                         [ <EXPLICIT_ROUTE> ]
                         <LABEL_REQUEST>
                         <CALL_OPS>
                         <CALL_ID>
                         [ <PROTECTION> ]
                         [ <LABEL_SET> ... ]
                         [ <SESSION_ATTRIBUTE> ]
                         [ <NOTIFY_REQUEST> ]
```

```
                          [ <ADMIN_STATUS> ]
                          <GENERALIZED_UNI>
                          [ <POLICY_DATA> ... ]
                          <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE>
                          <SENDER_TSPEC>
                          [ <ADSPEC> ]
                          [ <RECORD_ROUTE> ]
                          [ <SUGGESTED_LABEL> ]
                          [ <RECOVERY_LABEL> ]
```

The format of the sender description for bidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE>
                          <SENDER_TSPEC>
                          [ <ADSPEC> ]
                          [ <RECORD_ROUTE> ]
                          [ <SUGGESTED_LABEL> ]
                          [ <RECOVERY_LABEL> ]
                          <UPSTREAM_LABEL>
```

B.2.2 Resv

```
<Resv Message> ::=        <Common Header>
                          [ <INTEGRITY> ]
                          [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                          [ <MESSAGE_ID> ]
                          <SESSION>
                          <IF_ID_RSVP_HOP>
                          <TIME_VALUES>
                          <CALL_OPS>
                          <CALL_ID>
                          [ <RESV_CONFIRM> ]
                          [ <SCOPE> ]
                          [ <NOTIFY_REQUEST> ]
                          [ <ADMIN_STATUS> ]
                          [ <POLICY_DATA> ... ]
                          <STYLE>
                          <flow descriptor list>
```

```
    <flow descriptor list> ::= <FF flow descriptor list>
                             | <SE flow descriptor>
```

```
    <FF flow descriptor list> ::= <FLOWSPEC>
                                  <FILTER_SPEC>
                                  <LABEL>
                                  [ <RECORD_ROUTE> ]
                             | <FF flow descriptor list>
```

```
                              <FF flow descriptor>


      <FF flow descriptor> ::= [ <FLOWSPEC> ]
                               <FILTER_SPEC>
                               <LABEL>
                               [ <RECORD_ROUTE> ]


      <SE flow descriptor> ::= <FLOWSPEC>
                               <SE filter spec list>


      <SE filter spec list> ::= <SE filter spec>
                               | <SE filter spec list> <SE filter spec>


      <SE filter spec> ::=     <FILTER_SPEC>
                               <LABEL>
                               [ <RECORD_ROUTE> ]
```

B.2.3 PathTear

```
            <PathTear Message> ::= <Common Header>
                                   [ <INTEGRITY> ]
                                   <SESSION>
                                   <IF_ID_RSVP_HOP>
                                   <CALL_OPS>
                                   <CALL_ID>
                                   [ <sender descriptor> ]


             <sender descriptor> ::= (see earlier definition)
```

B.2.4 ResvTear

```
             <ResvTear Message> ::= <Common Header>
                                    [<INTEGRITY>]
                                    <SESSION>
                                    <IF_ID_RSVP_HOP>
                                    [ <SCOPE> ]
                                    <STYLE>
                                    <flow descriptor list>


             <flow descriptor list> ::= (see earlier definition)
```

B.2.5 PathErr

```
<PathErr message> ::= <Common Header>
                      [ <INTEGRITY> ]
                      <SESSION>
                      <ERROR_SPEC>
                      [ <POLICY_DATA> ...]
                      <CALL_OPS>
                      <CALL_ID>
                      [ <sender descriptor> ]
```

```
<sender descriptor> ::= (see earlier definition)
```

B.2.6 ResvErr

```
<ResvErr Message> ::=    <Common Header>
                         [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION>
                         <IF_ID_RSVP_HOP>
                         <ERROR_SPEC>
                         [ <SCOPE> ]
                         [ <ACCEPTABLE_LABEL_SET> ... ]
                         [ <POLICY_DATA> ... ]
                         <STYLE>
                         <error flow descriptor>
```

B.2.7 ResvConf

```
<ResvConf message> ::= <Common Header>
                       [ <INTEGRITY> ]
                       <SESSION>
                       <ERROR_SPEC>
                       <RESV_CONFIRM>
                       <STYLE>
                       <flow descriptor list>
```

```
<flow descriptor list> ::= (see earlier definition)
```

B.2.8 Notify

```
<Notify message>  ::= <Common Header>
                      [<INTEGRITY>]
                      [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                      [ <MESSAGE_ID> ]
```

```
                                <ERROR_SPEC>
                                <notify session list>
```

```
      <notify session list> ::= [ <notify session list> ]
                                  <upstream notify session> |
                                  <downstream notify session>
```

```
   <upstream notify session> ::= <SESSION>
                                 <CALL_ID>
                                 [ <ADMIN_STATUS> ]
                                 [<POLICY_DATA>...]
                                 <sender descriptor>
```

```
  <downstream notify session> ::= <SESSION>
                                  <CALL_ID>
                                  [<POLICY_DATA>...]
                                  <flow descriptor list>
```

B.2.9 Hello

```
      <Hello Message> ::= <Common Header>
                          [ <INTEGRITY> ]
                          <HELLO>
                          [ <RESTART_CAP> ]
```

B.2.10 ACK/NACK

```
       <ACK Message> ::= <Common Header>
                         [ <INTEGRITY> ]
                         <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
```

B.2.11 Srefresh

```
      <Srefresh Message> ::= <Common Header>
                             [ <INTEGRITY> ]
                             [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                             [ <MESSAGE_ID> ]
                             <srefresh list> | <source srefresh list>
```

```
         <srefresh list> ::= <MESSAGE_ID LIST> | <MESSAGE_ID MCAST_LIST>
                               [ <srefresh list> ]
```

```
<source srefresh list> ::= <MESSAGE_ID SRC_LIST>
                                [ <source srefresh list> ]
```

B.3 RSVP Objects

B.3.1 SESSION Class (C-num = 1)

B.3.1.1 *IPv4/UDP SESSION object: Class = 1, C-Type = 1*

```
+-------------+-------------+-------------+-------------+
|             IPv4 DestAddress (4 bytes)                |
+-------------+-------------+-------------+-------------+
| Protocol Id |    Flags    |          DstPort          |
+-------------+-------------+-------------+-------------+
```

- DestAddress
  - The IPv4 unicast or multicast destination address of the session. This field must be non-zero.
- Protocol Id
  - The IP Protocol Identifier for the data flow. This field must be non-zero.
- Flags
  - 0x01 = E_Police flag (The E_Police flag is used in Path messages to determine the effective "edge" of the network, to control traffic policing. If the sender host is not itself capable of traffic policing, it will set this bit on in Path messages it sends. The first node whose RSVP is capable of traffic policing will do so (if appropriate to the service) and turn the flag off.)
- DstPort
  - The UDP/TCP destination port for the session. Zero may be used to indicate `none'.

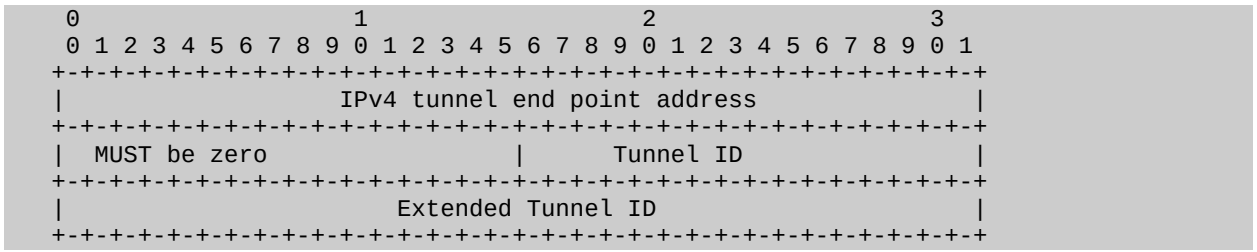
B.3.1.2 *IPv6/UDP SESSION object: Class = 1, C-Type = 2*

```
+-------------+-------------+-------------+-------------+
|                                                       |
+                                                       +
|                                                       |
+               IPv6 DestAddress (16 bytes)             +
|                                                       |
+                                                       +
|                                                       |
+-------------+-------------+-------------+-------------+
| Protocol Id |     Flags   |          DstPort          |
+-------------+-------------+-------------+-------------+
```

- DestAddress
  - The IPv6 unicast or multicast destination address of the session. This field must be non-zero.

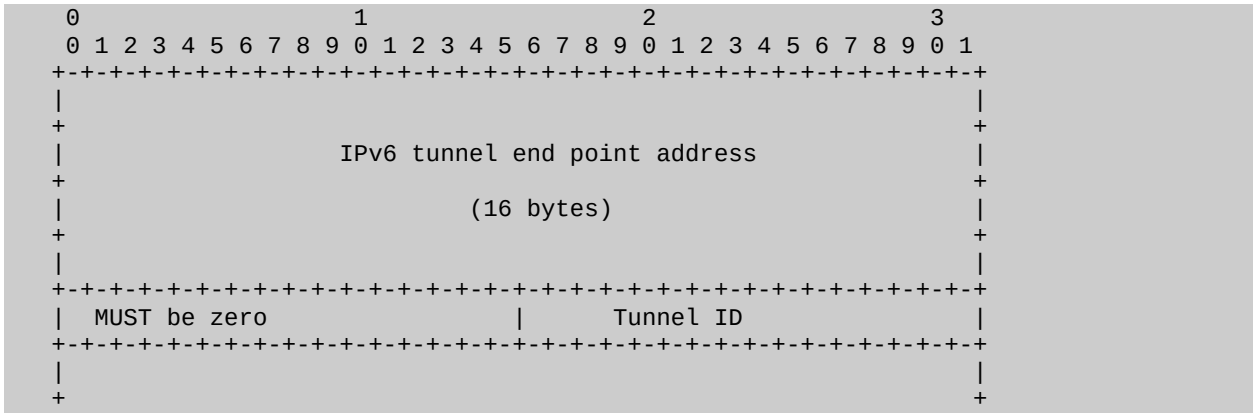
- Protocol Id
  - The IP Protocol Identifier for the data flow. This field must be non-zero.
- Flags
  - 0x01 = E_Police flag (The E_Police flag is used in Path messages to determine the effective "edge" of the network, to control traffic policing. If the sender host is not itself capable of traffic policing, it will set this bit on in Path messages it sends. The first node whose RSVP is capable of traffic policing will do so (if appropriate to the service) and turn the flag off.)
- DstPort
  - The UDP/TCP destination port for the session. Zero may be used to indicate `none'.

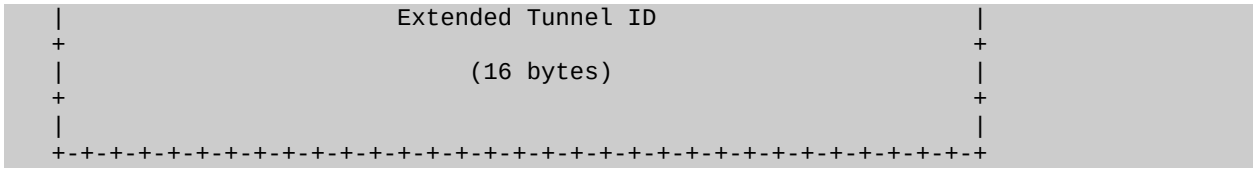
B.3.1.3 *LSP_TUNNEL_IPv4 SESSION object: Class = 1, C-Type = 7*



- IPv4 tunnel end point address
  - IPv4 address of the egress node for the tunnel.
- Tunnel ID
  - A 16-bit identifier used in the SESSION that remains constant over the life of the tunnel.
- Extended Tunnel ID
  - A 32-bit identifier used in the SESSION that remains constant over the life of the tunnel. Normally set to all zeros. Ingress nodes that wish to narrow the scope of a SESSION to the ingress-egress pair may place their IPv4 address here as a globally unique identifier.

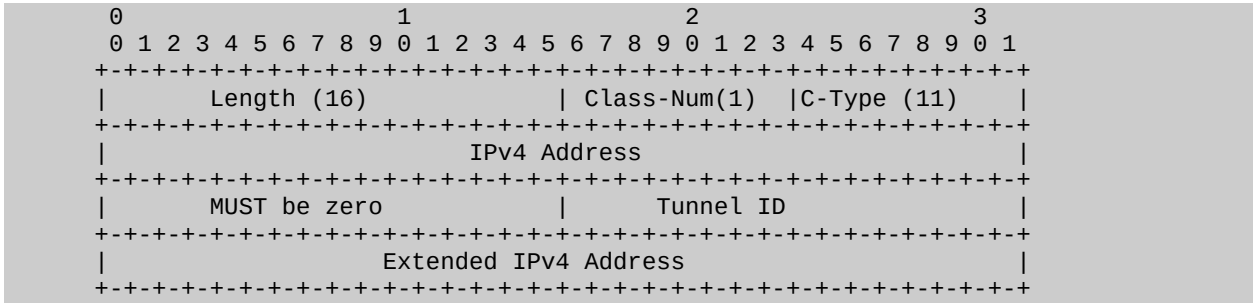
B.3.1.4 *LSP_TUNNEL_IPv6 SESSION object: Class = 1, C-Type = 8*





- IPv6 tunnel end point address
  - IPv6 address of the egress node for the tunnel.
- Tunnel ID
  - A 16-bit identifier used in the SESSION that remains constant over the life of the tunnel.
- Extended Tunnel ID
  - A 16-byte identifier used in the SESSION that remains constant over the life of the tunnel. Normally set to all zeros. Ingress nodes that wish to narrow the scope of a SESSION to the ingress-egress pair may place their IPv6 address here as a globally unique identifier.

B.3.1.5 *UNI_IPv4_SESSION object: Class = 1, C-Type = 11*

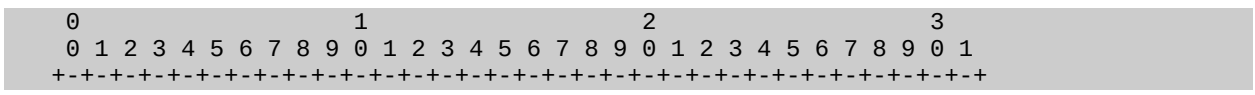


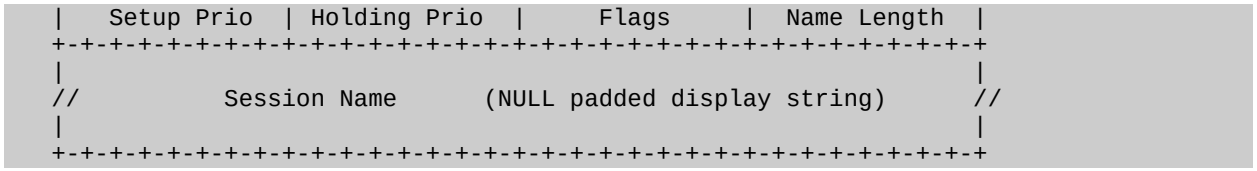
The C-Type value (11) will distinguish UNI-related RSVP Sessions from other RSVP sessions.

B.3.2 SESSION ATTRIBUTE Class (C-num = 207)

The Session Attribute Class is 207. Two C_Types are defined, LSP_TUNNEL, C-Type = 7 and LSP_TUNNEL_RA, C-Type = 1. The LSP_TUNNEL_RA C-Type includes all the same fields as the LSP_TUNNEL C-Type. Additionally it carries resource affinity information. The formats are as follows:

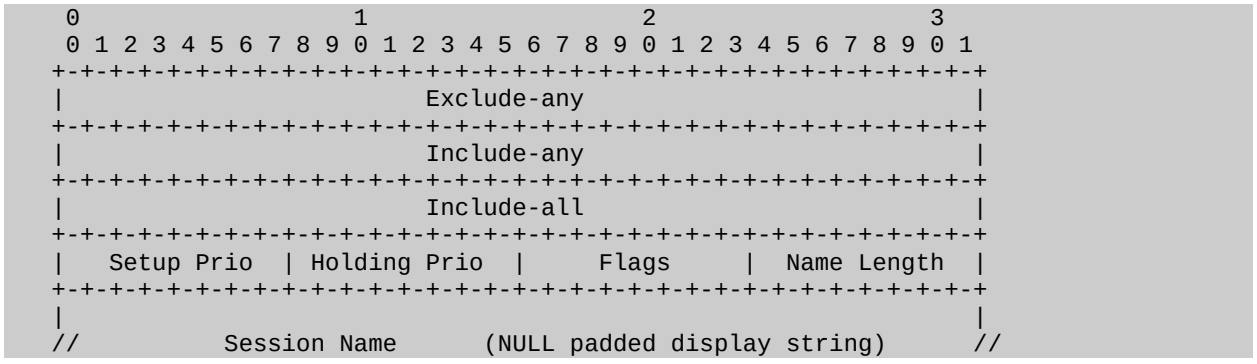
B.3.2.1 *LSP_TUNNEL SESSION_ATTRIBUTE objrct: class = 207, C-Type = 7 (Format without resource affinities)*

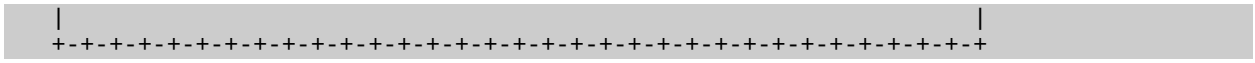




- Setup Priority
  - The priority of the session with respect to taking resources, in the range of 0 to 7. The value 0 is the highest priority. The Setup Priority is used in deciding whether this session can preempt another session.
- Holding Priority
  - The priority of the session with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session.
- Flags
  - 0x01 Local protection desired
    - This flag permits transit routers to use a local repair mechanism which may result in violation of the explicit route object. When a fault is detected on an adjacent downstream link or node, a transit router can reroute traffic for fast service restoration.
  - 0x02 Label recording desired
    - This flag indicates that label information should be included when doing a route record.
  - 0x04 SE Style desired
    - This flag indicates that the tunnel ingress node may choose to reroute this tunnel without tearing it down. A tunnel egress node SHOULD use the SE Style when responding with a Resv message.
- Name Length
  - The length of the display string before padding, in bytes.
- Session Name
  - A null padded string of characters.

B.3.2.2 *LSP_TUNNEL_RA SESSION_ATTRIBUTE object: class = 207, C-Type = 1 (Format with resource affinities)*





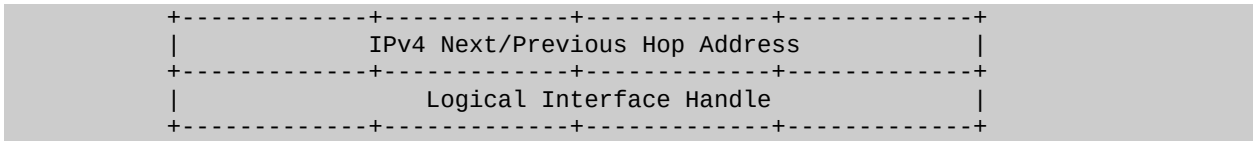
- Exclude-any
  - A 32-bit vector representing a set of attribute filters associated with a tunnel any of which renders a link unacceptable.
- Include-any
  - A 32-bit vector representing a set of attribute filters associated with a tunnel any of which renders a link acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.
- Include-all
  - A 32-bit vector representing a set of attribute filters associated with a tunnel all of which must be present for a link to be acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.
- Setup Priority
  - The priority of the session with respect to taking resources, in the range of 0 to 7. The value 0 is the highest priority. The Setup Priority is used in deciding whether this session can preempt another session.
- Holding Priority
  - The priority of the session with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session.
- Flags
  - 0x01 Local protection desired
    - This flag permits transit routers to use a local repair mechanism which may result in violation of the explicit route object. When a fault is detected on an adjacent downstream link or node, a transit router can reroute traffic for fast service restoration.
  - 0x02 Label recording desired
    - This flag indicates that label information should be included when doing a route record.
  - 0x04 SE Style desired
    - This flag indicates that the tunnel ingress node may choose to reroute this tunnel without tearing it down. A tunnel egress node SHOULD use the SE Style when responding with a Resv message.
- Name Length
  - The length of the display string before padding, in bytes.
- Session Name
  - A null padded string of characters.

B.3.3 RSVP_HOP Class (C-num = 3)

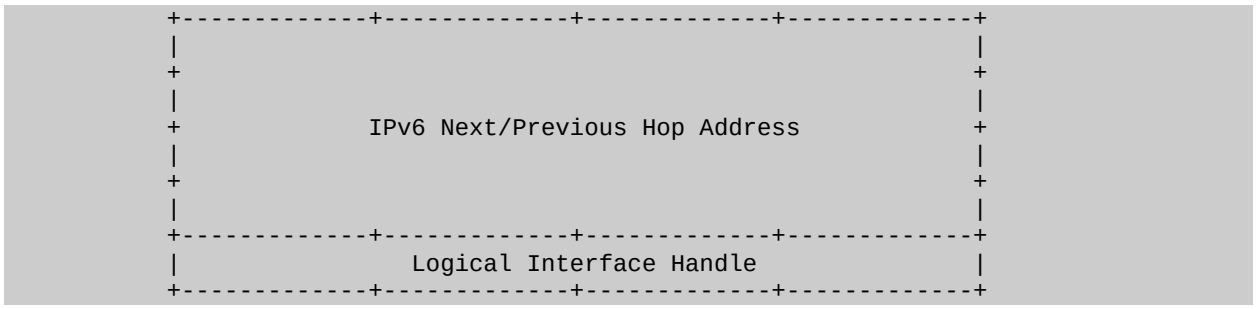
This object carries the IP address of the interface through which the last RSVP-knowledgeable hop forwarded this message. The Logical Interface Handle (LIH) is used to distinguish logical outgoing interfaces, as discussed in Sections 3.3 and 3.9. A node receiving an LIH in a Path message saves its value and returns it in

the HOP objects of subsequent Resv messages sent to the node that originated the LIH. The LIH should be identically zero if there is no logical interface handle.

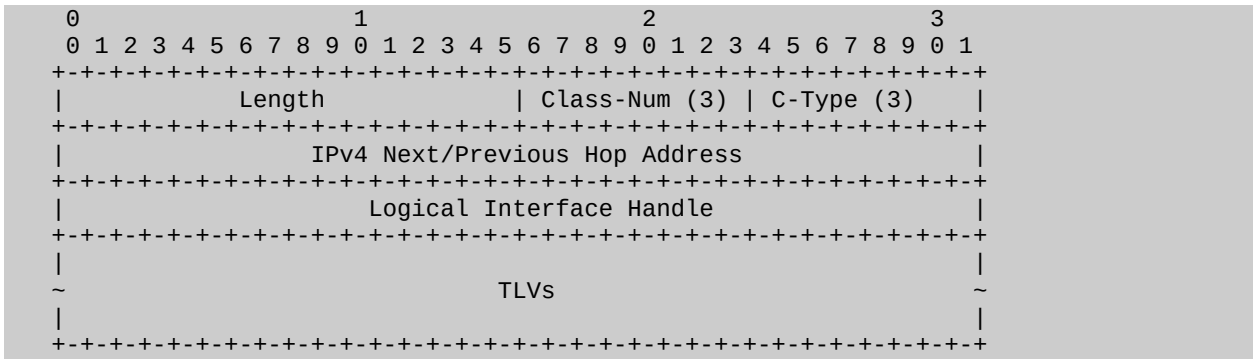
B.3.3.1 *IPv4 RSVP_HOP object: Class = 3, C-Type = 1*



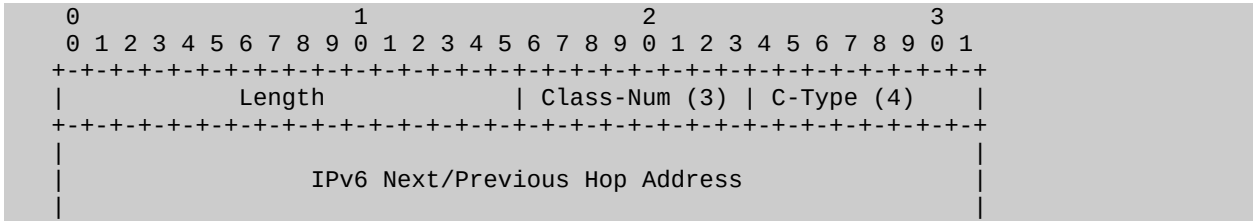
B.3.3.2 *IPv6 RSVP_HOP object: Class = 3, C-Type = 2*

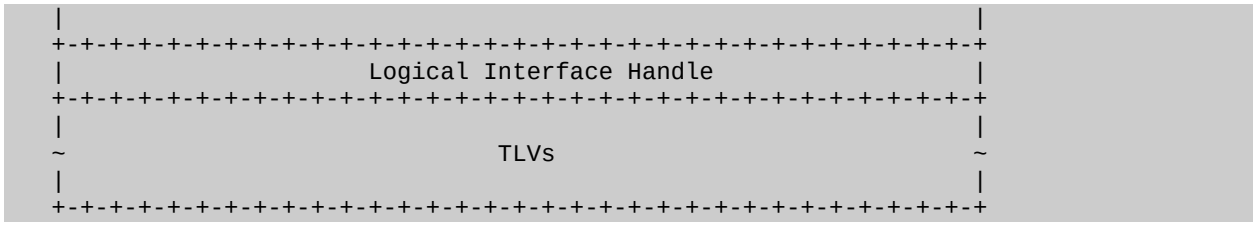


B.3.3.3 *IPv4 IF_ID RSVP_HOP object: Class = 3, C-Type = 3*



B.3.3.4 *IPv6 IF_ID RSVP_HOP object: Class = 3, C-Type = 4*



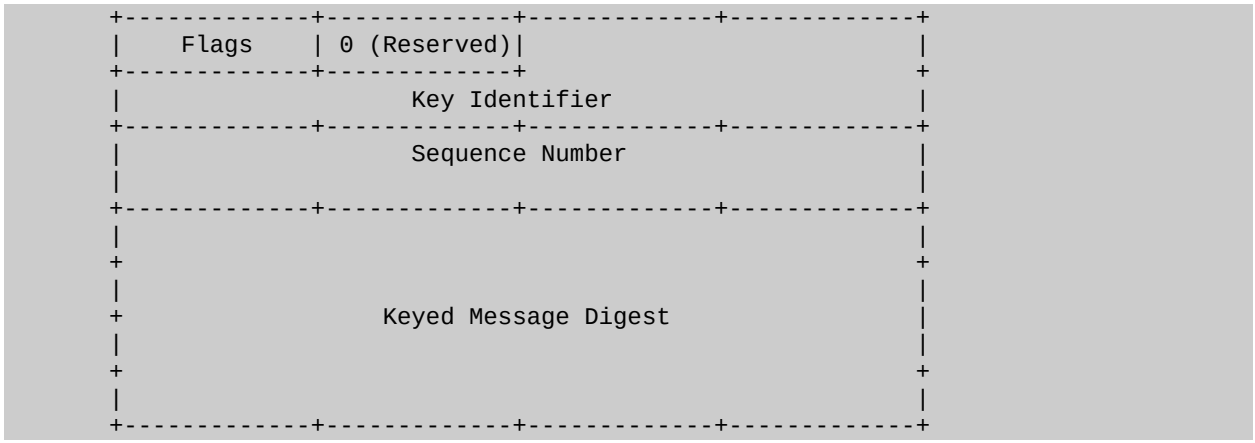


The Logical Interface Handle (LIH) is used to distinguish logical outgoing interfaces. A node receiving an LIH in a Path message saves its value and returns it in the HOP objects of subsequent Resv messages sent to the node that originated the LIH. The LIH should be identically zero if there is no logical interface handle.

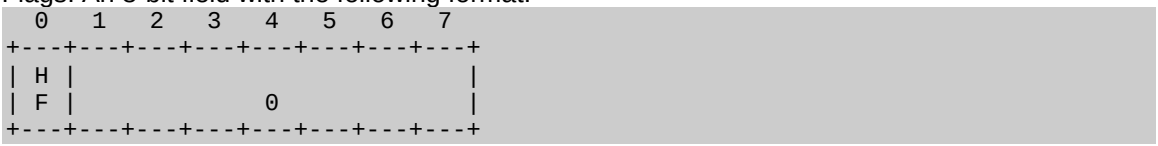
TLVs are used to identify the data channel(s) associated with an LSP. For a unidirectional LSP, a downstream data channel MUST be indicated. For bidirectional LSPs, a common downstream and upstream data channel is normally indicated. In the special case where a bidirectional LSP that traverses a bundled link, it is possible to specify a downstream data channel that differs from the upstream data channel. Data channels are specified from the viewpoint of the sender of the Path message. The IF_ID RSVP_HOP object SHOULD NOT be used when no TLVs are needed.

B.3.4 INTEGRITY Class (C-num = 4)

B.3.4.1 *Keyed Message Digest INTEGRITY object: Class = 4, C-Type = 1*



- Flags: An 8-bit field with the following format:



- Bit 0: Handshake Flag (HF) concerns the integrity handshake mechanism (Section 4.3). Message senders willing to respond to integrity handshake messages SHOULD set this flag to 1 whereas those that will reject integrity handshake messages SHOULD set this to 0.

- Key Identifier: An unsigned 48-bit number that MUST be unique for a given sender. Locally unique Key Identifiers can be generated using some combination of the address (IP or MAC or LIH) of the sending interface and the key number. The combination of the Key Identifier and the sending system's IP address uniquely identifies the security association.
- Sequence Number: An unsigned 64-bit monotonically increasing, unique sequence number. Sequence Number values may be any monotonically increasing sequence that provides the INTEGRITY object [of each RSVP message] with a tag that is unique for the associated key's lifetime.
- Keyed Message Digest: The digest MUST be a multiple of 4 octets long. For HMAC-MD5, it will be 16 bytes long.

B.3.5 TIME_VALUES Class (C-num = 5)

B.3.5.1 *TIME_VALUES object: Class = 5, C-Type = 1*

```
+-------------+-------------+-------------+-------------+
|                  Refresh Period R                     |
+-------------+-------------+-------------+-------------+
```

- Refresh Period
  - The refresh timeout period R used to generate this message; in milliseconds.

B.3.6 ERROR_SPEC Class (C-num = 6)

B.3.6.1 *IPv4 ERROR_SPEC object: Class = 6, C-Type = 1*

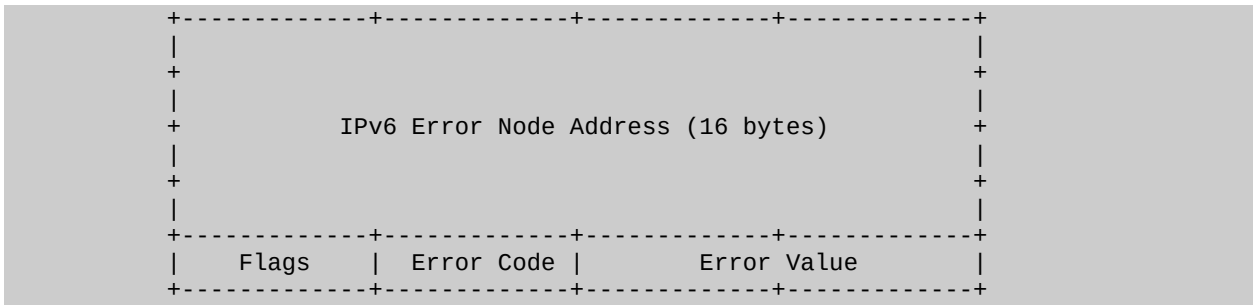
```
+-------------+-------------+-------------+-------------+
|            IPv4 Error Node Address (4 bytes)          |
+-------------+-------------+-------------+-------------+
|    Flags    |  Error Code |        Error Value        |
+-------------+-------------+-------------+-------------+
```

- Error Node Address
  - The IPv4 address of the node in which the error was detected.
- Flags
  - 0x01 = InPlace
    - This flag is used only for an ERROR_SPEC object in a ResvErr message. If it on, this flag indicates that there was, and still is, a reservation in place at the failure point.

- 0x02 = NotGuilty
  - This flag is used only for an ERROR_SPEC object in a ResvErr message, and it is only set in the interface to the receiver application. If it on, this flag indicates that the FLOWSPEC that failed was strictly greater than the FLOWSPEC requested by this receiver.

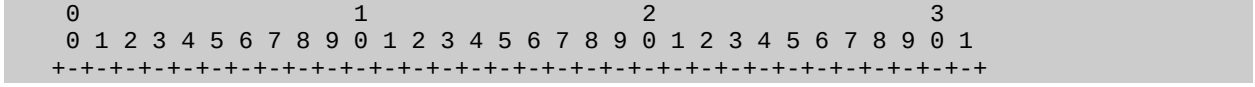
- Error Code
  - A one-octet error description.
- Error Value
  - A two-octet field containing additional information about the error. Its contents depend upon the Error Type.

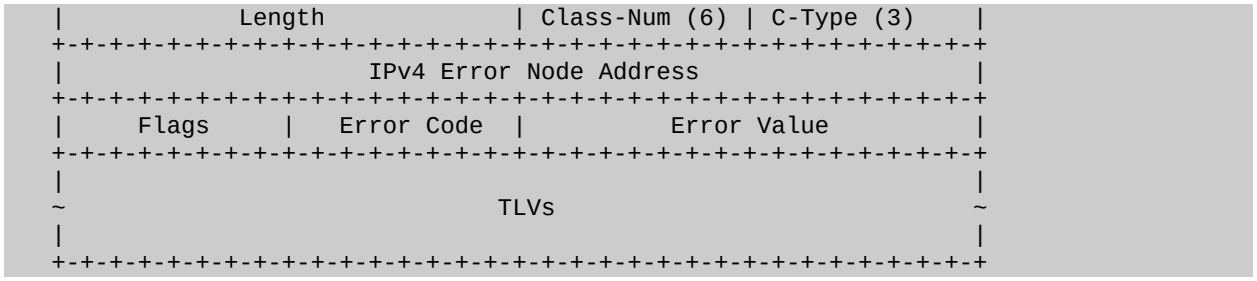
B.3.6.2 *IPv6 ERROR_SPEC object: Class = 6, C-Type = 2*



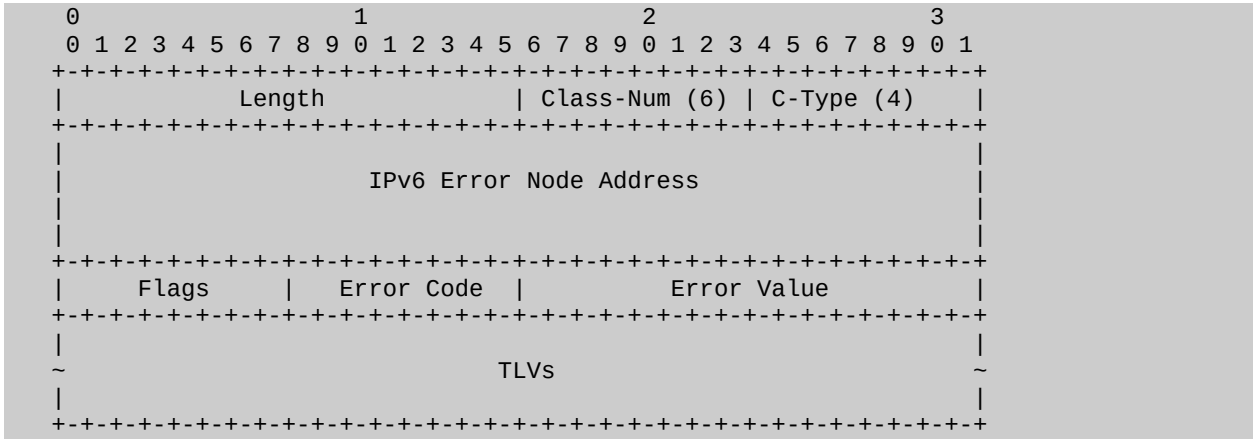
- Error Node Address
  - The IPv6 address of the node in which the error was detected.
- Flags
  - 0x01 = InPlace
    - This flag is used only for an ERROR_SPEC object in a ResvErr message. If it on, this flag indicates that there was, and still is, a reservation in place at the failure point.
  - 0x02 = NotGuilty
    - This flag is used only for an ERROR_SPEC object in a ResvErr message, and it is only set in the interface to the receiver application. If it on, this flag indicates that the FLOWSPEC that failed was strictly greater than the FLOWSPEC requested by this receiver.
- Error Code
  - A one-octet error description.
- Error Value
  - A two-octet field containing additional information about the error. Its contents depend upon the Error Type.

B.3.6.3 *IPv4 IF_ID ERROR_SPEC objects: Class = 6, C-Type = 3*





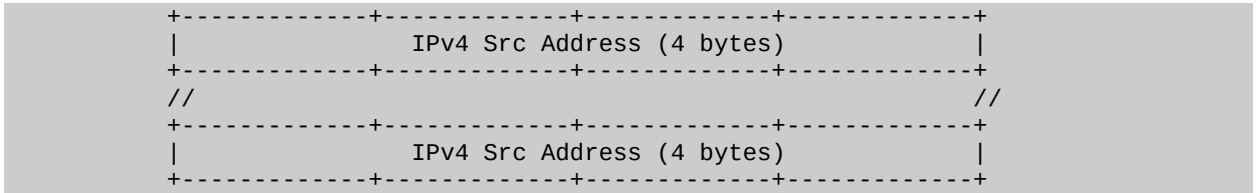
B.3.6.4 *IPv6 IF_ID ERROR_SPEC objects: Class = 6, C-Type = 4*



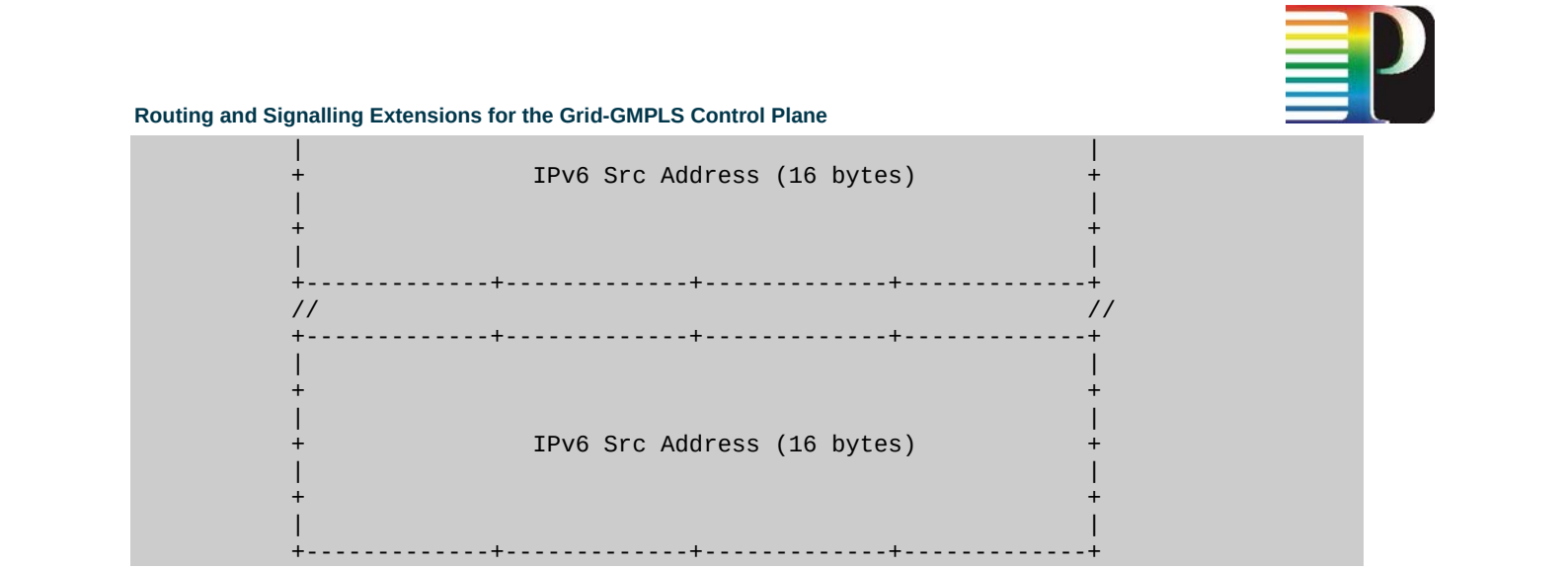
B.3.7 SCOPE Class (C-num = 7)

This object contains a list of IP addresses, used for routing messages with wildcard scope without loops. The addresses must be listed in ascending numerical order.

B.3.7.1 *IPv4 SCOPE List object: Class = 7, C-Type = 1*



B.3.7.2 *IPv6 SCOPE List object: Class = 7, C-Type = 2*



```
          |                                                               |
          +                IPv6 Src Address (16 bytes)                    +
          |                                                               |
          +                                                               +
          |                                                               |
          +-------------+-------------+-------------+-------------+
          //                                                              //
          +-------------+-------------+-------------+-------------+
          |                                                               |
          +                                                               +
          |                                                               |
          +                IPv6 Src Address (16 bytes)                    +
          |                                                               |
          +                                                               +
          |                                                               |
          +-------------+-------------+-------------+-------------+
```

B.3.8 STYLE Class (C-num = 8)

B.3.8.1 *STYLE object: Class = 8, C-Type = 1*

```
          +-------------+-------------+-------------+-------------+
          |   Flags     |              Option Vector                |
          +-------------+-------------+-------------+-------------+
```

- Flags: 8 bits
  - (None assigned yet)
- Option Vector: 24 bits
  - A set of bit fields giving values for the reservation options. If new options are added in the future, corresponding fields in the option vector will be assigned from the least-significant end. If a node does not recognize a style ID, it may interpret as much of the option vector as it can, ignoring new fields that may have been defined. The option vector bits are assigned (from the left) as follows:
    - 19 bits: Reserved
    - 2 bits: Sharing control
      00b: Reserved
      01b: Distinct reservations
      10b: Shared reservations
      11b: Reserved
    - 3 bits: Sender selection control
      000b: Reserved
      001b: Wildcard
      010b: Explicit
      011b - 111b: Reserved

The low order bits of the option vector are determined by the style, as follows: WF 10001b, FF 01010b, SE 10010b

B.3.9 FLOWSPEC Class (C-num = 9)

B.3.9.1 *Reserved (obsolete) flowspec object: Class = 9, C-Type = 1*

B.3.9.2 *IntServ FLOWSPEC object: Class = 9, C-Type = 2*

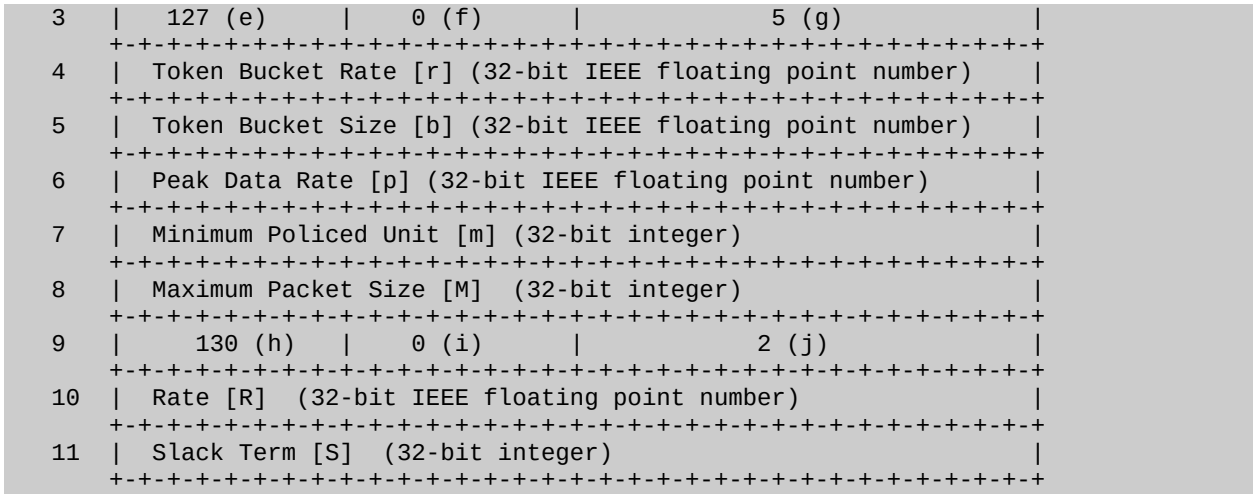
FLOWSPEC object when requesting Controlled-Load service

```
      31           24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
1    | 0 (a) |    reserved           |             7 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
2    |    5  (c)     |0| reserved    |             6 (d)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
3    |   127 (e)     |    0 (f)      |             5 (g)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
4    |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
5    |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
6    |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
7    |  Minimum Policed Unit [m] (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
8    |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- (a) - Message format version number (0)
- (b) - Overall length (7 words not including header)
- (c) - Service header, service number 5 (Controlled-Load)
- (d) - Length of controlled-load data, 6 words not including per-service header
- (e) - Parameter ID, parameter 127 (Token Bucket TSpec)
- (f) - Parameter 127 flags (none set)
- (g) - Parameter 127 length, 5 words not including per-service header

FLOWSPEC Object when Requesting Guaranteed Service

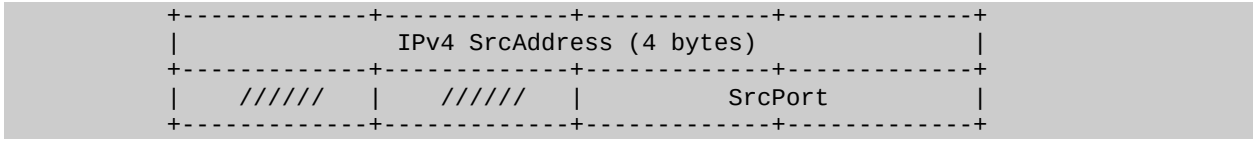
```
      31           24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
1    | 0 (a) |    Unused             |            10 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
2    |    2  (c)     |0| reserved    |             9 (d)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



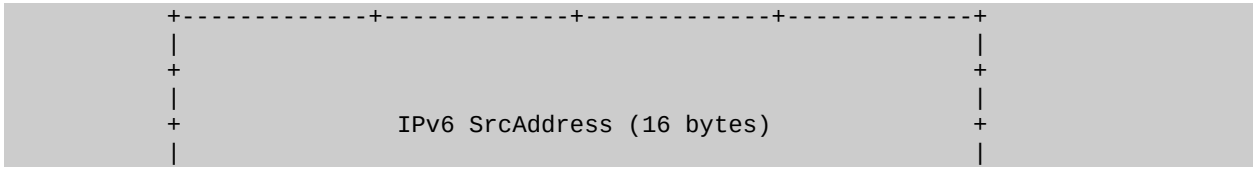
- (a) - Message format version number (0)
- (b) - Overall length (9 words not including header)
- (c) - Service header, service number 2 (Guaranteed)
- (d) - Length of per-service data, 9 words not including per-service header
- (e) - Parameter ID, parameter 127 (Token Bucket TSpec)
- (f) - Parameter 127 flags (none set)
- (g) - Parameter 127 length, 5 words not including parameter header
- (h) - Parameter ID, parameter 130 (Guaranteed Service RSpec)
- (i) - Parameter 130 flags (none set)
- (j) - Parameter 130 length, 2 words not including parameter header

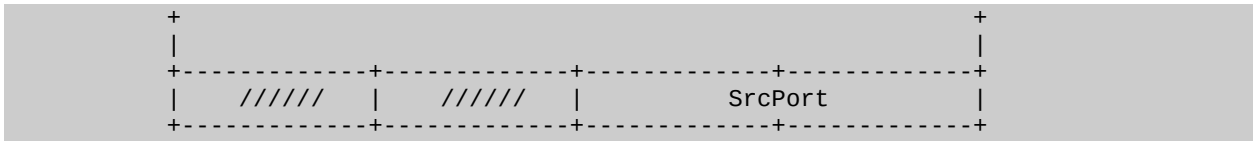
B.3.10 FILTER_SPEC Class (C-num = 10)

B.3.10.1 *IPv4 FILTER_SPEC object: Class = 10, C-Type = 1*

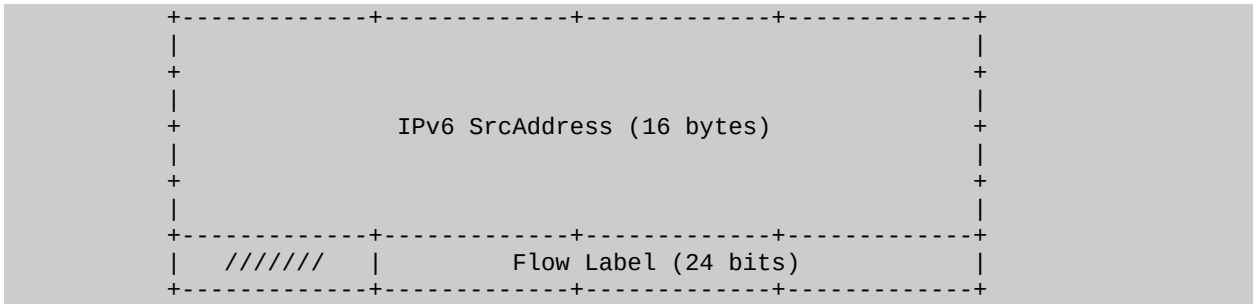


B.3.10.2 *IPv6 FILTER_SPEC object: Class = 10, C-Type = 2*





B.3.10.3 *IPv6 Flow-label FILTER_SPEC object: Class = 10, C-Type = 3*



- SrcAddress
  - The IP source address for a sender host. Must be non-zero.
- SrcPort
  - The UDP/TCP source port for a sender, or zero to indicate `none'.
- Flow Label
  - A 24-bit Flow Label, defined in IPv6. This value may be used by the packet classifier to efficiently identify the packets belonging to a particular (sender->destination) data flow.

B.3.10.4 *LSP_TUNNEL_IPv4 FILTER_SPEC object: Class = 10, C-Type = 7*

The format of the LSP_TUNNEL_IPv4 FILTER_SPEC object is identical to the LSP_TUNNEL_IPv4 SENDER_TEMPLATE object.

B.3.10.5 *LSP_TUNNEL_IPv6 FILTER_SPEC object: Class = 10, C-Type = 8*

The format of the LSP_TUNNEL_IPv6 FILTER_SPEC object is identical to the LSP_TUNNEL_IPv6 SENDER_TEMPLATE object.

B.3.11 SENDER_TEMPLATE Class (C-num = 11)

B.3.11.1 *IPv4 SENDER_TEMPLATE object: Class = 11, C-Type = 1*

Definition same as IPv4/UDP FILTER_SPEC object.

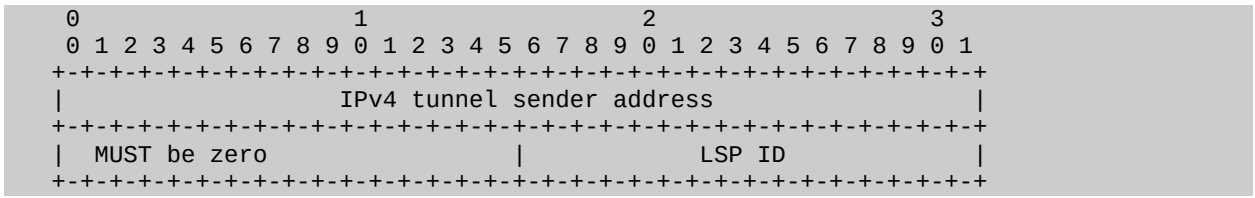
B.3.11.2 *IPv6 SENDER_TEMPLATE object: Class = 11, C-Type = 2*

Definition same as IPv6/UDP FILTER_SPEC object.

B.3.11.3 *IPv6 Flow-label SENDER_TEMPLATE object: Class = 11, C-Type = 3*

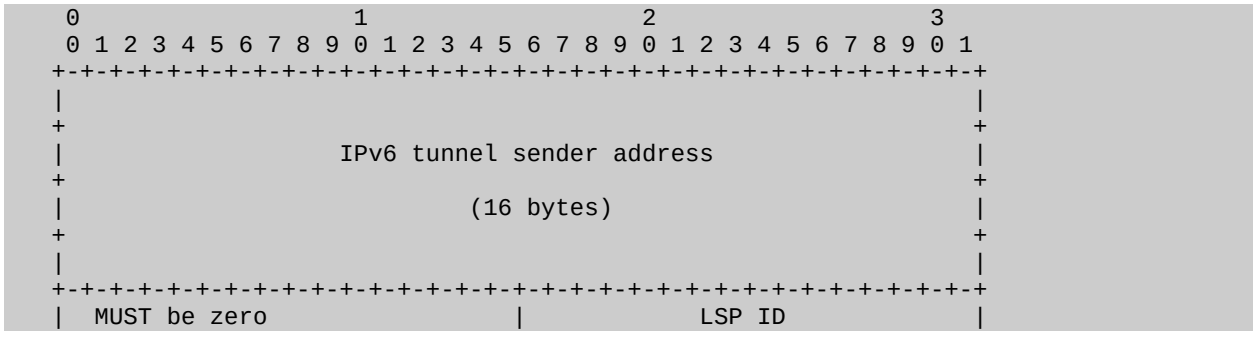
Definition same as IPv6 Flow-label FILTER_SPEC object.

B.3.11.4 *LSP_TUNNEL_IPv4 SENDER_TEMPLATE object: Class = 11, C-Type = 7*



- IPv4 tunnel sender address
  - IPv4 address for a sender node
- LSP ID
  - A 16-bit identifier used in the SENDER_TEMPLATE and the FILTER_SPEC that can be changed to allow a sender to share resources with itself.

B.3.11.5 *LSP_TUNNEL_IPv6 SENDER_TEMPLATE object: Class = 11, C-Type = 8*



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- IPv6 tunnel sender address
  - IPv6 address for a sender node
- LSP ID
  - A 16-bit identifier used in the SENDER_TEMPLATE and the FILTER_SPEC that can be changed to allow a sender to share resources with itself.

B.3.12 SENDER_TSPEC Class (C-num = 12)

B.3.12.1 *Intserv SENDER_TSPEC object: Class = 12, C-Type = 2*

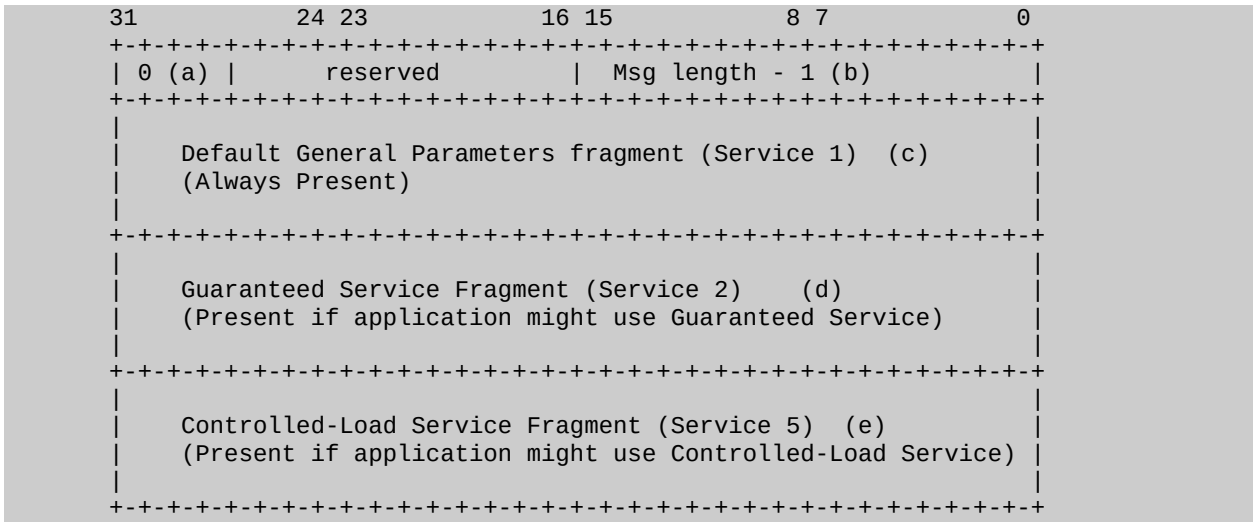
This TSpec carries traffic information usable by either the Guaranteed or Controlled-Load QoS control services.

```
      31           24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
1    | 0 (a) |    reserved           |             7 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
2    |    1  (c)     |0| reserved    |             6 (d)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
3    |   127 (e)     |    0 (f)      |             5 (g)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
4    |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
5    |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
6    |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
7    |  Minimum Policed Unit [m] (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
8    |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- (a) - Message format version number (0)
- (b) - Overall length (7 words not including header)
- (c) - Service header, service number 1 (default/global information)
- (d) - Length of service 1 data, 6 words not including header
- (e) - Parameter ID, parameter 127 (Token_Bucket_TSpec)
- (f) - Parameter 127 flags (none set)
- (g) - Parameter 127 length, 5 words not including header

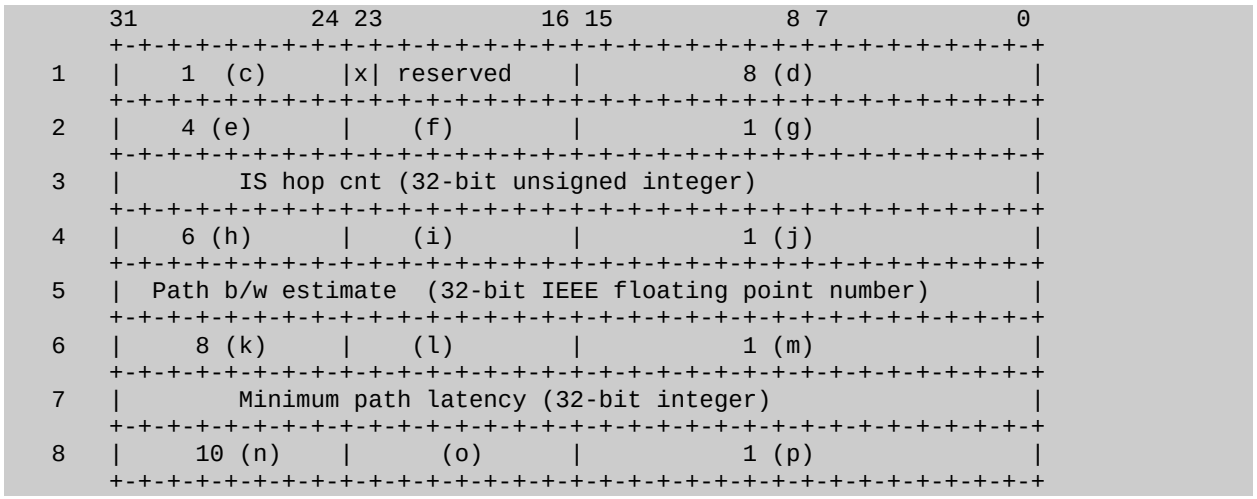
B.3.13 ADSPEC Class (C-num = 13)

B.3.13.1 *Intserv ADSPEC object: Class = 13, C-Type = 2*



- (a) - Message format version number (0)
- (b) - Overall message length not including header word
- (c, d, e) - Data fragments

All RSVP ADSPECs carry the general characterization parameters defined in [RFC 2215]. Values for global or default general parameters (values which apply to the all services or the path itself) are carried in the per-service data fragment for service number 1, as shown in the picture above. This fragment is always present, and always first in the message.



```
   9   |      Composed MTU (32-bit unsigned integer)                   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- (c) - Per-Service header, service number 1 (Default General Parameters)
- (d) - Global Break bit ([RFC 2215], Parameter 2) (marked x) and length of General Parameters data block.
- (e) - Parameter ID, parameter 4 (Number-of-IS-hops param from [RFC 2215])
- (f) - Parameter 4 flag byte
- (g) - Parameter 4 length, 1 word not including header
- (h) - Parameter ID, parameter 6 (Path-BW param from [RFC 2215])
- (i) - Parameter 6 flag byte
- (j) - Parameter 6 length, 1 word not including header
- (k) - Parameter ID, parameter 8 (minimum path latency from [RFC 2215])
- (l) - Parameter 8 flag byte
- (m) - Parameter 8 length, 1 word not including header
- (n) - Parameter ID, parameter 10 (composed path MTU from [RFC 2215])
- (o) - Parameter 10 flag byte
- (p) - Parameter 10 length, 1 word not including header

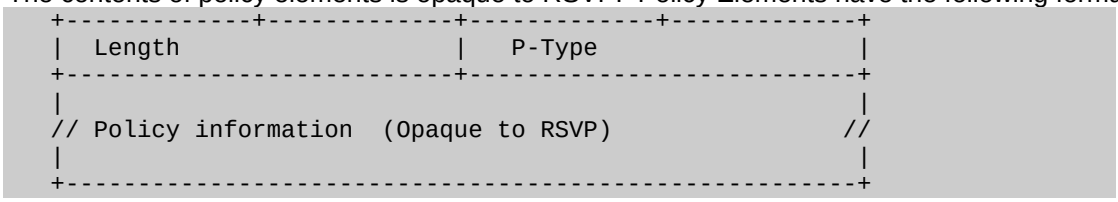
B.3.14 POLICY_DATA Class (C-num = 14)

B.3.14.1 *Type 1 POLICY_DATA object: Class = 14, C-Type = 1*

```
       +-------------+-------------+-------------+-------------+
       |  Length                   | POLICY_DATA |      1      |
       +---------------------------+-------------+-------------+
       |  Data Offset              | 0 (reserved)              |
       +---------------------------+-------------+-------------+
       |                                                       |
       // Option List                                        //
       |                                                       |
       +-------------------------------------------------------+
       |                                                       |
       // Policy Element List                                 //
       |                                                       |
       +-------------------------------------------------------+
```

- Data Offset: 16 bits
  - The offset in bytes of the data portion (from the first byte of the object header).
- Reserved: 16 bits
  - Always 0.
- Option List: Variable length
  - The list of options and their usage is defined in RFC 2750.
- Policy Element List: Variable length

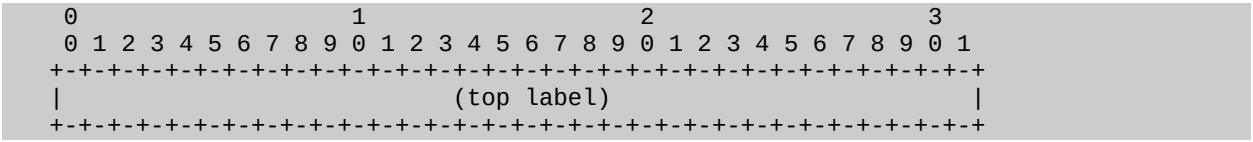
- The contents of policy elements is opaque to RSVP. Policy Elements have the following format:



B.3.15 LABEL Class (C-num = 16)

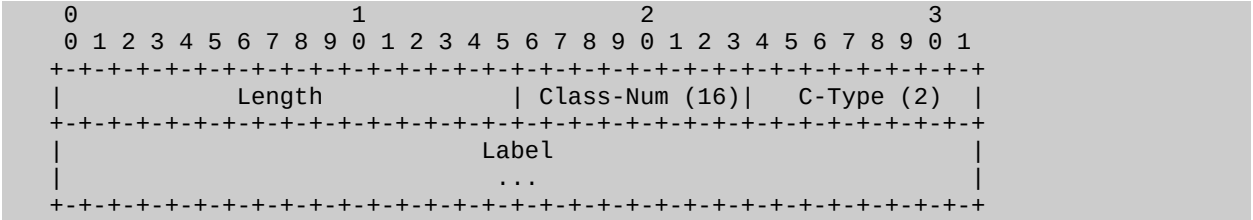
Labels MAY be carried in Resv messages. For the FF and SE styles, a label is associated with each sender. The label for a sender MUST immediately follow the FILTER_SPEC for that sender in the Resv message.

B.3.15.1 *LABEL object: Class = 16, C_Type = 1*

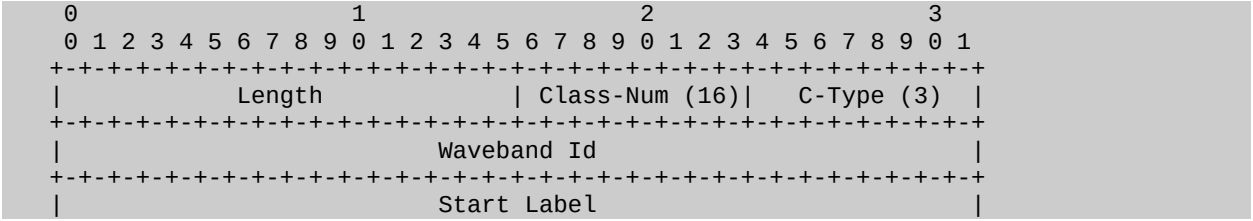


The contents of a LABEL is a single label, encoded in 4 octets. Each generic MPLS label is an unsigned integer in the range 0 through 1048575. Generic MPLS labels and FR labels are encoded right aligned in 4 octets. ATM labels are encoded with the VPI right justified in bits 0-15 and the VCI right justified in bits 16-31.

B.3.15.2 *Generalized LABEL object: Class = 16, C_Type = 2*



B.3.15.3 *Waveband Switching object: Class = 16, C_Type = 3*



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          End Label                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

B.3.16 LABEL REQUEST Class (C-num = 19)

B.3.16.1 *LABEL REQUEST without Label Range: Class = 19, C_Type = 1*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |             L3PID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Reserved
  - This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt.
- L3PID
  - an identifier of the layer 3 protocol using this path. Standard Ethertype values are used.

B.3.16.2 *LABEL REQUEST with ATM Label Range: Class = 19, C_Type = 2*

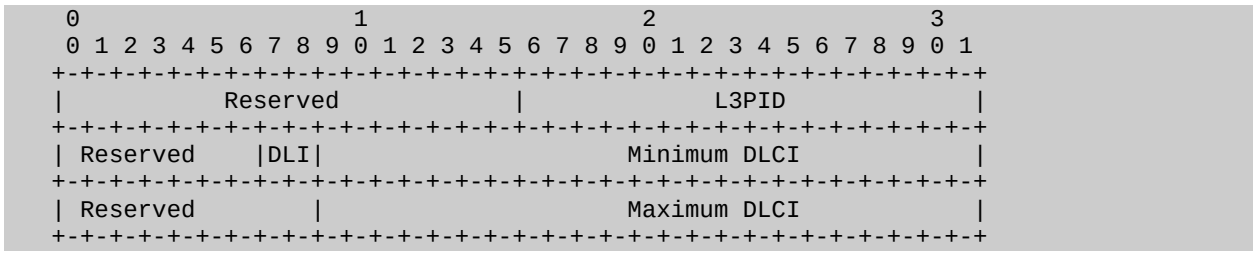
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |             L3PID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M| Res |    Minimum VPI         |      Minimum VCI              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Res  |    Maximum VPI         |      Maximum VCI              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Reserved (Res)
  - This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt.
- L3PID
  - an identifier of the layer 3 protocol using this path. Standard Ethertype values are used.
- M
  - Setting this bit to one indicates that the node is capable of merging in the data plane
- Minimum VPI (12 bits)
  - This 12 bit field specifies the lower bound of a block of Virtual Path Identifiers that is supported on the originating switch. If the VPI is less than 12-bits it MUST be right justified in this field and preceding bits MUST be set to zero.
- Minimum VCI (16 bits)

- This 16 bit field specifies the lower bound of a block of Virtual Connection Identifiers that is supported on the originating switch. If the VCI is less than 16-bits it MUST be right justified in this field and preceding bits MUST be set to zero.

- Maximum VPI (12 bits)
  - This 12 bit field specifies the upper bound of a block of Virtual Path Identifiers that is supported on the originating switch. If the VPI is less than 12-bits it MUST be right justified in this field and preceding bits MUST be set to zero.
- Maximum VCI (16 bits)
  - This 16 bit field specifies the upper bound of a block of Virtual Connection Identifiers that is supported on the originating switch. If the VCI is less than 16-bits it MUST be right justified in this field and preceding bits MUST be set to zero.

B.3.16.3 *LABEL REQUEST with Frame Relay Label Range: Class = 19, C_Type = 3*



- Reserved
  - This field is reserved. It MUST be set to zero on transmission and ignored on receipt.
- L3PID
  - an identifier of the layer 3 protocol using this path. Standard Ethertype values are used.
- DLI
  - DLCI Length Indicator. The number of bits in the DLCI. The following values are supported:

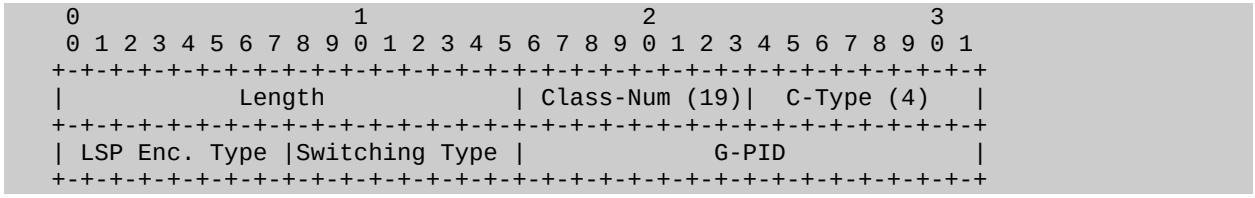
Len	DLCI bits
0	10
2	23

- Minimum DLCI
  - This 23-bit field specifies the lower bound of a block of Data Link Connection Identifiers (DLCIs) that is supported on the originating switch. The DLCI MUST be right justified in this field and unused bits MUST be set to 0.
- Maximum DLCI
  - This 23-bit field specifies the upper bound of a block of Data Link Connection Identifiers (DLCIs) that is supported on the originating switch. The DLCI MUST be right justified in this field and unused bits MUST be set to 0.

B.3.16.4 *GENERALIZED LABEL REQUEST object: Class = 19, C-Type = 4*

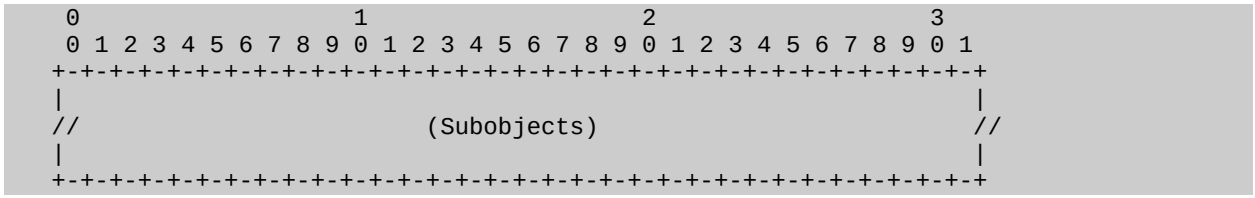
A Path message SHOULD contain as specific an LSP (Label Switched Path) Encoding Type as possible to allow the maximum flexibility in switching by transit LSRs. A Generalized Label Request object is set by the ingress node, transparently passed by transit nodes, and used by the egress node. The Switching Type field may also be updated hop-by-hop.

The format of a Generalized Label Request object is:

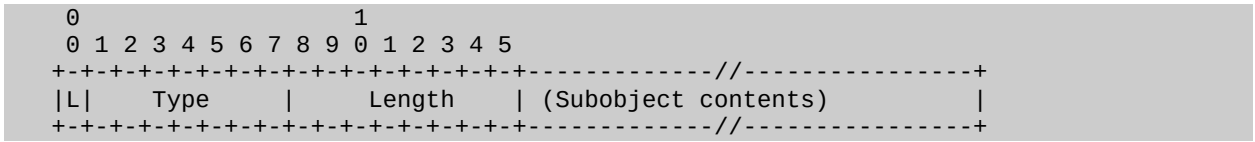


B.3.17 EXPLICIT ROUTE Class (C-num = 20)

B.3.17.1 *EXPLICIT_ROUTE object: Class = 20, C_Type = 1*



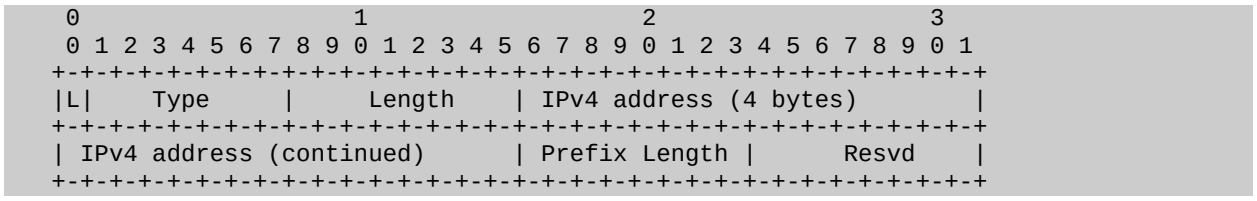
The contents of an EXPLICIT_ROUTE object are a series of variable-length data items called subobjects. Each subobject has the form:



- L
  - The L bit is an attribute of the subobject. The L bit is set if the subobject represents a loose hop in the explicit route. If the bit is not set, the subobject represents a strict hop in the explicit route.
- Type
  - The Type indicates the type of contents of the subobject. Currently defined values are:
    1 IPv4 prefix
    2 IPv6 prefix

32 Autonomous system number

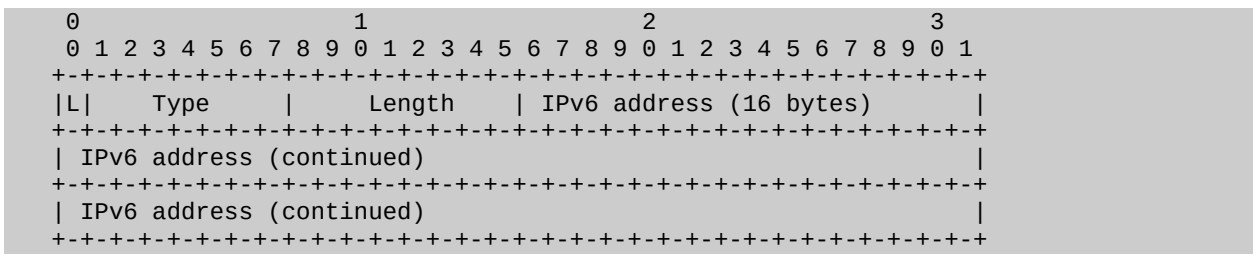
Subobject 1: IPv4 prefix



- L
  - The L bit is an attribute of the subobject. The L bit is set if the subobject represents a loose hop in the explicit route. If the bit is not set, the subobject represents a strict hop in the explicit route.
- Type
  - 0x01 IPv4 address
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always 8.
- IPv4 address
  - An IPv4 address. This address is treated as a prefix based on the prefix length value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.
- Prefix length
  - Length in bits of the IPv4 prefix
- Padding
  - Zero on transmission. Ignored on receipt.

The contents of an IPv4 prefix subobject are a 4-octet IPv4 address, a 1-octet prefix length, and a 1-octet pad. The abstract node represented by this subobject is the set of nodes that have an IP address which lies within this prefix. Note that a prefix length of 32 indicates a single IPv4 node.

Subobject 2: IPv6 Prefix



```
   | IPv6 address (continued)                                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv6 address (continued)      | Prefix Length |      Resvd    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- L
  - The L bit is an attribute of the subobject. The L bit is set if the subobject represents a loose hop in the explicit route. If the bit is not set, the subobject represents a strict hop in the explicit route.
- Type
  - 0x02 IPv6 address
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always 20.
- IPv6 address
  - An IPv6 address. This address is treated as a prefix based on the prefix length value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.
- Prefix Length
  - Length in bits of the IPv6 prefix.
- Padding
  - Zero on transmission. Ignored on receipt.

The contents of an IPv6 prefix subobject are a 16-octet IPv6 address, a 1-octet prefix length, and a 1-octet pad. The abstract node represented by this subobject is the set of nodes that have an IP address which lies within this prefix. Note that a prefix length of 128 indicates a single IPv6 node.

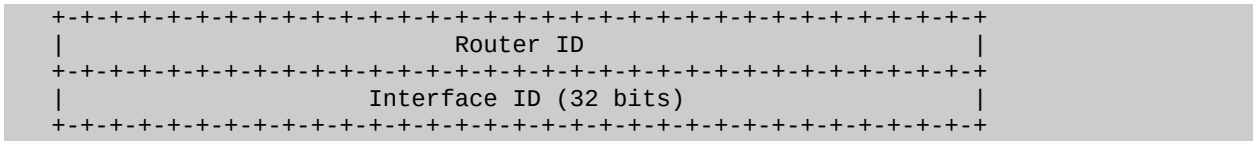
Subobject 32: Autonomous System Number

The contents of an Autonomous System (AS) number subobject are a 2- octet AS number. The abstract node represented by this subobject is the set of nodes belonging to the autonomous system. The length of the AS number subobject is 4 octets.

Subobject 4: Unnumbered Interface ID

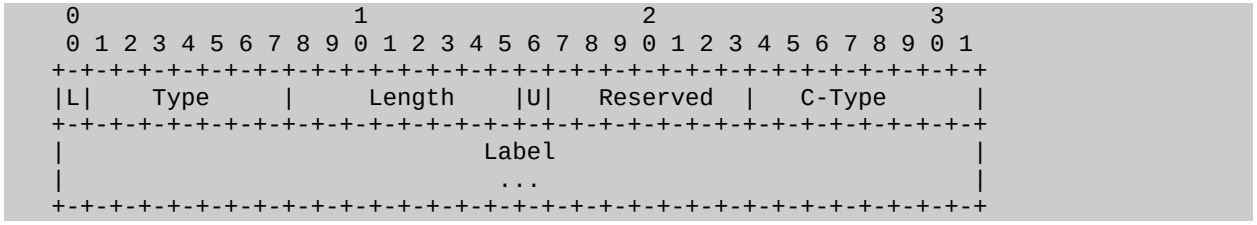
A new subobject of the Explicit Route Object (ERO) is used to specify unnumbered links. This subobject has the following format:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    |    Reserved (MUST be zero)    |
```



- Type is 4 (Unnumbered Interface ID). The Length is 12.
- The Interface ID is the identifier assigned to the link by the LSR specified by the router ID.

Subobject 4: Label

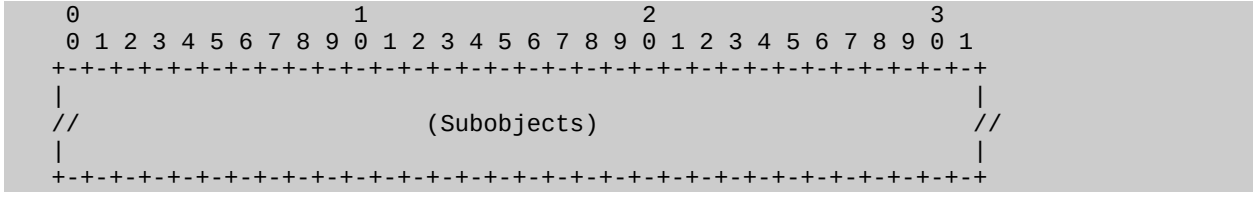


See [RFC3471] for a description of L, U and Label parameters.

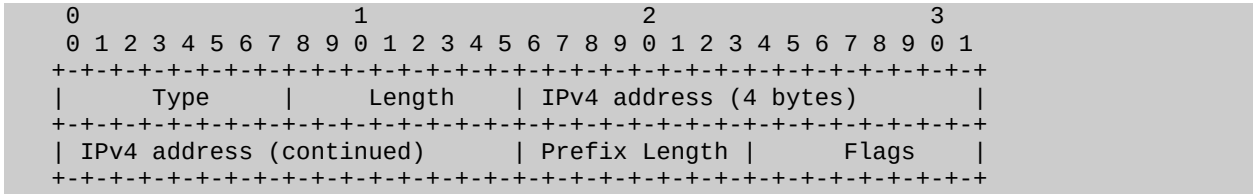
- Type
  - 3 Label
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always divisible by 4.
- C-Type
  - The C-Type of the included Label Object. Copied from the Label
- Object.

B.3.18 RECORD ROUTE Class (C-num = 21)

B.3.18.1 *RECORD_ROUTE object: Class = 20, C_Type = 1*

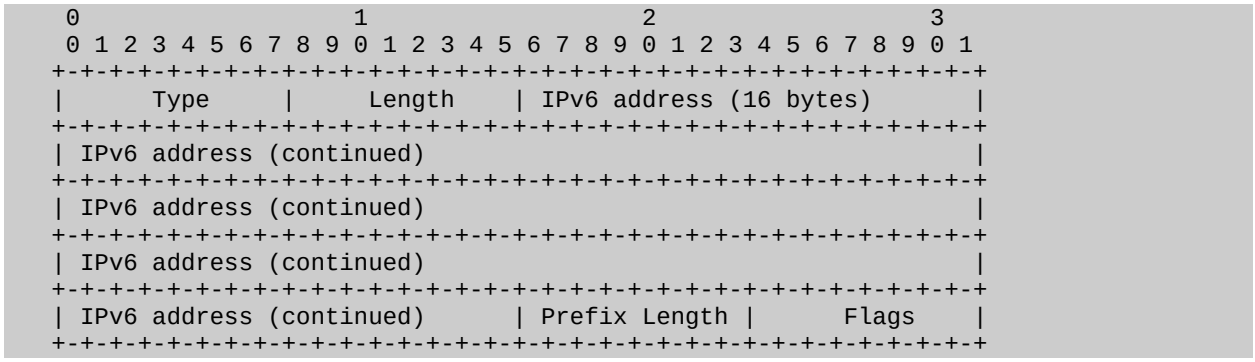


Subobject 1: IPv4 address



- Type
  - 0x01 IPv4 address
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always 8.
- IPv4 address
  - A 32-bit unicast, host address. Any network-reachable interface address is allowed here. Illegal addresses, such as certain loopback addresses, SHOULD NOT be used.
- Prefix length
  - 32
- Flags
  - 0x01 Local protection available
    - Indicates that the link downstream of this node is protected via a local repair mechanism. This flag can only be set if the Local protection flag was set in the SESSION_ATTRIBUTE object of the corresponding Path message.
  - 0x02 Local protection in use
    - Indicates that a local repair mechanism is in use to maintain this tunnel (usually in the face of an outage of the link it was previously routed over).

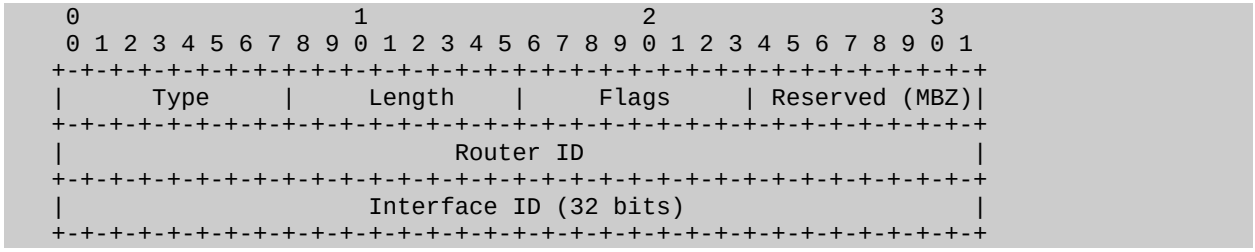
Subobject 2: IPv6 address



- Type

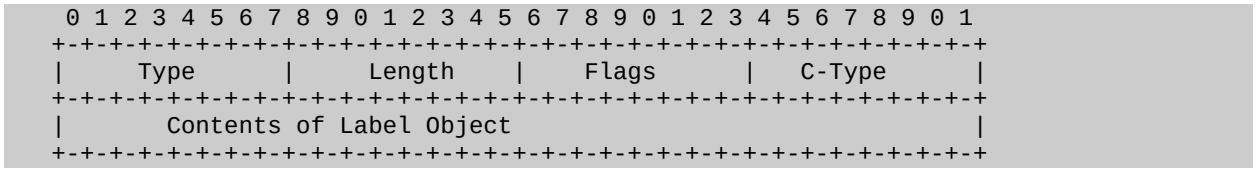
- 0x02 IPv6 address
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always 20.
- IPv6 address
  - A 128-bit unicast host address.
- Prefix length
  - 128
- Flags
  - 0x01 Local protection available
    - Indicates that the link downstream of this node is protected via a local repair mechanism. This flag can only be set if the Local protection flag was set in the SESSION_ATTRIBUTE object of the corresponding Path message.
  - 0x02 Local protection in use
    - Indicates that a local repair mechanism is in use to maintain this tunnel (usually in the face of an outage of the link it was previously routed over).

Subobject 4: Unnumbered Interface ID



- Type is 4 (Unnumbered Interface ID);
- Length is 12.
- Flags are
  - 0x01 Local protection available
    - Indicates that the link downstream of this node is protected via a local repair mechanism. This flag can only be set if the Local protection flag was set in the SESSION_ATTRIBUTE object of the corresponding Path message.
  - 0x02 Local protection in use
    - Indicates that a local repair mechanism is in use to maintain this tunnel (usually in the face of an outage of the link it was previously routed over).

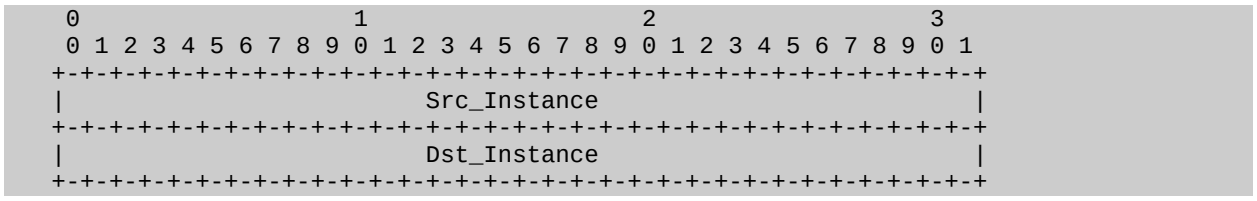
Subobject 3: Label



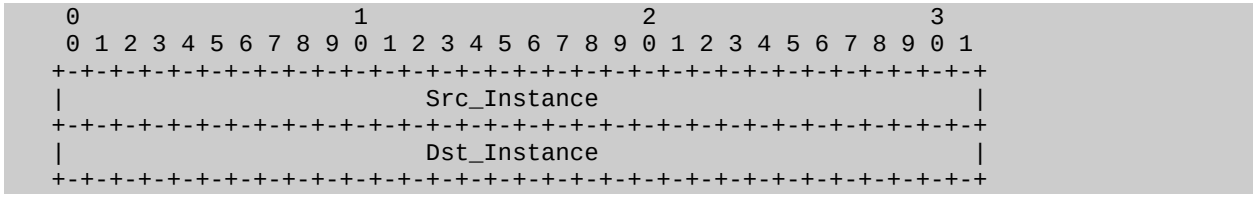
- Type
  - 0x03 Label
- Length
  - The Length contains the total length of the subobject in bytes, including the Type and Length fields.
- Flags
  - 0x01 = Global label
    - This flag indicates that the label will be understood if received on any interface.
- C-Type
  - The C-Type of the included Label Object. Copied from the Label Object.
- Contents of Label Object
  - The contents of the Label Object. Copied from the Label Object

B.3.19 HELLO Class (C-num = 22)

B.3.19.1 *HELLO REQUEST object: Class = 22, C_Type = 1*



B.3.19.2 *HELLO_ACK object: Class = 22, C_Type = 2*



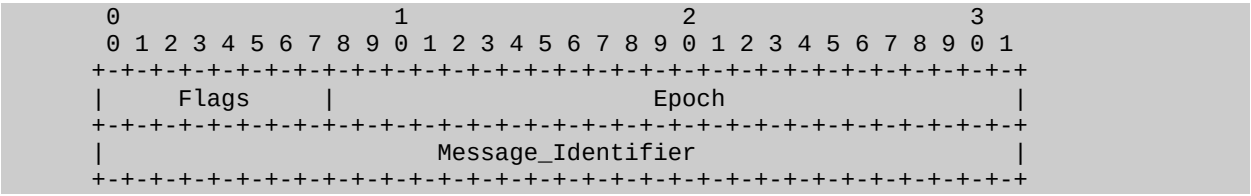
- Src_Instance: 32 bits
  - a 32 bit value that represents the sender's instance. The advertiser maintains a per neighbour representation/value. This value MUST change when the sender is reset, when the node reboots,

or when communication is lost to the neighbouring node and otherwise remains the same. This field MUST NOT be set to zero (0).

- Dst_Instance: 32 bits
  - The most recently received Src_Instance value received from the neighbour. This field MUST be set to zero (0) when no value has ever been seen from the neighbour.

B.3.20 MESSAGE_ID Class (C-num = 23)

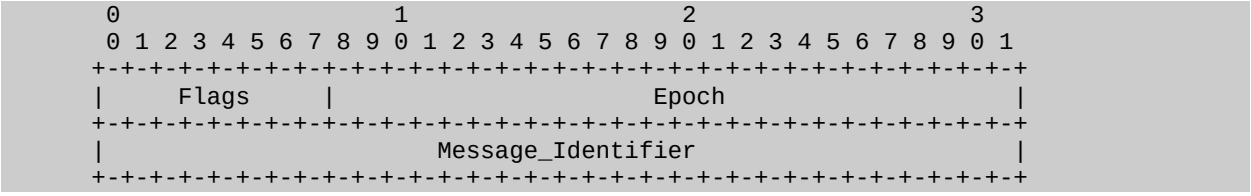
B.3.20.1 *MESSAGE_ID object: Class = 23, C_Type = 1*



- Flags: 8 bits
  - 0x01 = ACK_Desired flag
    - Indicates that the sender requests the receiver to send an acknowledgment for the message.
- Epoch: 24 bits
  - A value that indicates when the Message_Identifier sequence has reset. SHOULD be randomly generated each time a node reboots or the RSVP agent is restarted. The value SHOULD NOT be the same as was used when the node was last operational. This value MUST NOT be changed during normal operation.
- Message_Identifier: 32 bits
  - When combined with the message generator's IP address, the Message_Identifier field uniquely identifies a message. The values placed in this field change incrementally and only decrease when the Epoch changes or when the value wraps.

B.3.21 MESSAGE_ID_ACK Class (C-num = 24)

B.3.21.1 *MESSAGE_ID_ACK object: Class = 23, C_Type = 1*



- Flags: 8 bits

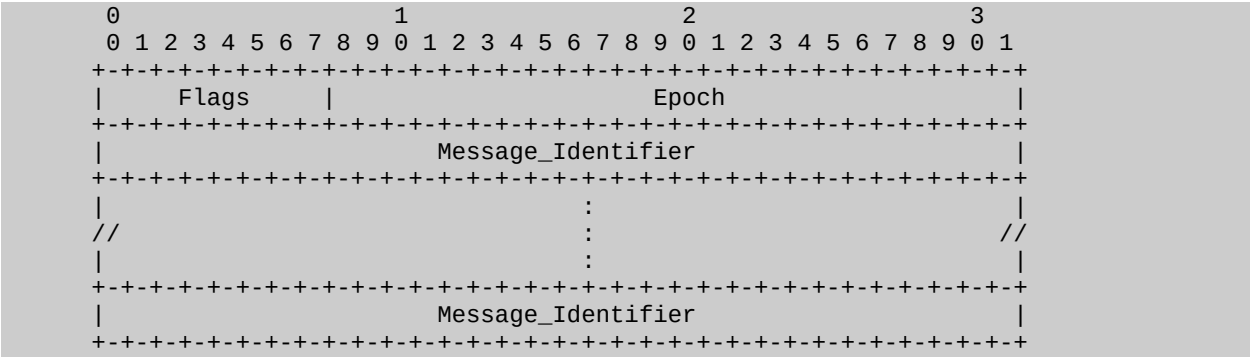
- No flags are currently defined. This field MUST be zero on transmission and ignored on receipt.
- Epoch: 24 bits
  - The Epoch field copied from the message being acknowledged.
- Message_Identifier: 32 bits
  - The Message_Identifier field copied from the message being acknowledged.

B.3.21.2 *MESSAGE_ID_NACK object: Class = 23, C_Type = 2*

Definition is the same as the MESSAGE_ID_ACK object.

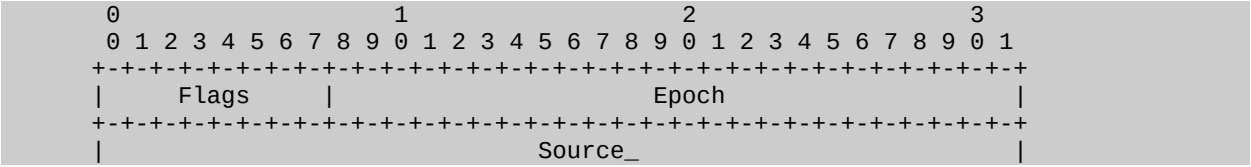
B.3.22 MESSAGE_ID_LIST Class (C-num = 25)

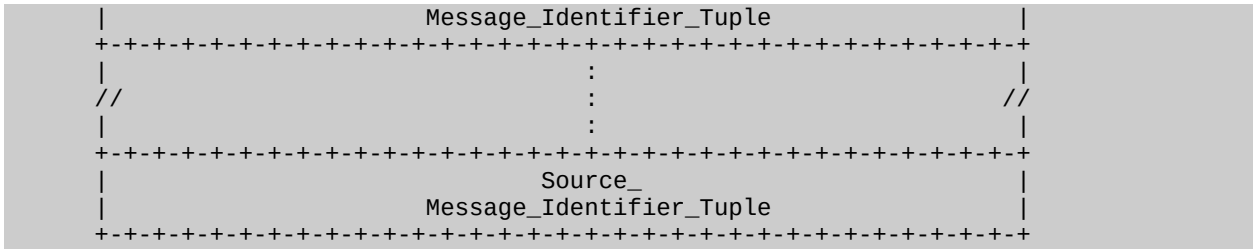
B.3.22.1 *MESSAGE_ID_LISTobject: Class = 25, C_Type = 1*



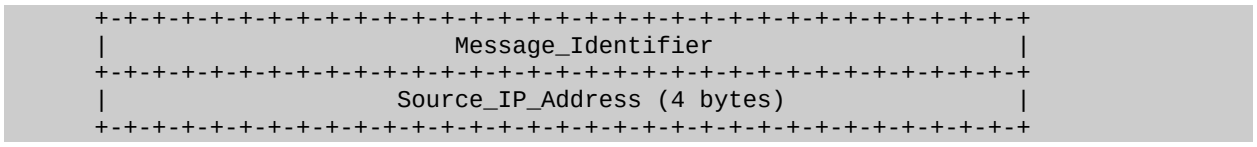
- Flags: 8 bits
  - No flags are currently defined. This field MUST be zero on transmission and ignored on receipt.
- Epoch: 24 bits
  - The Epoch field from the MESSAGE_ID object corresponding to the trigger message that advertised the state being refreshed.
- Message_Identifier: 32 bits
  - The Message_Identifier field from the MESSAGE_ID object corresponding to the trigger message that advertised the state being refreshed. One or more Message_Identifiers may be included.

B.3.22.2 *IPv4/MESSAGE_ID SRC_LIST object: Class = 25, C_Type = 2*

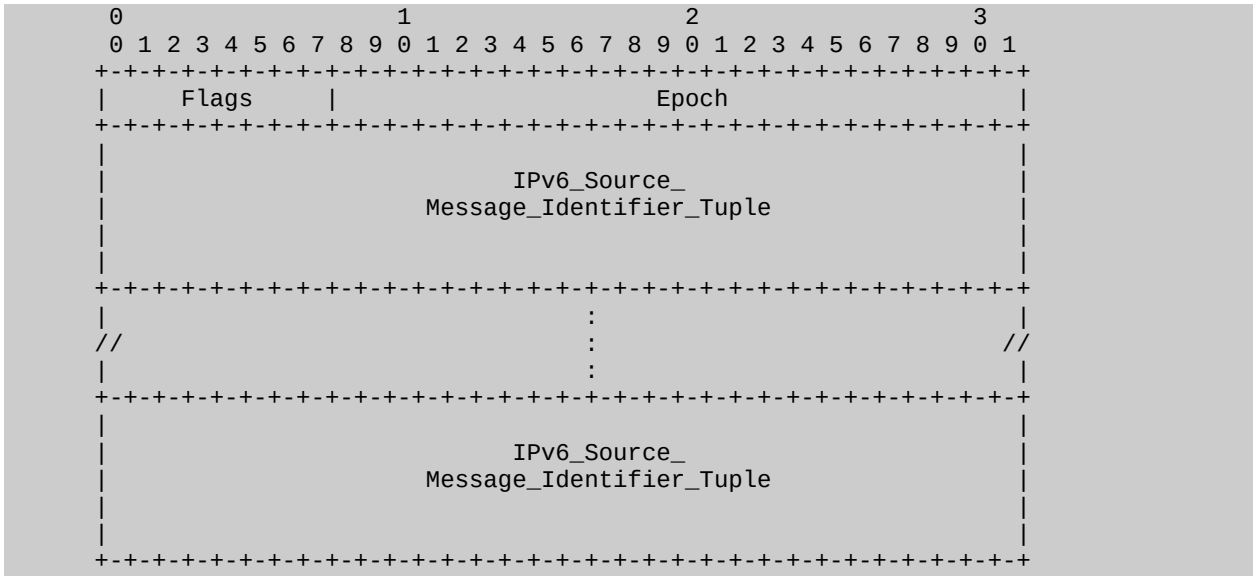




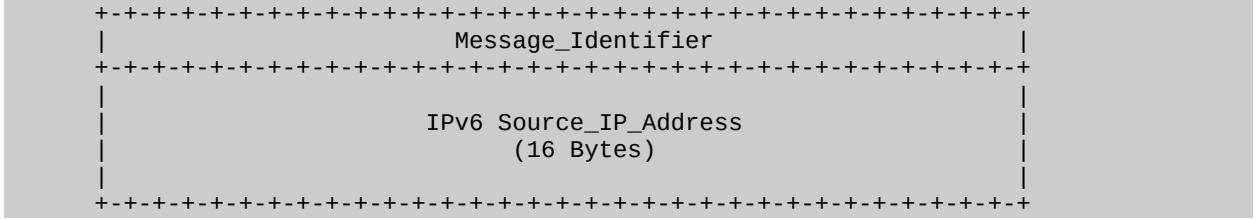
Where a Source_Message_Identifier_Tuple consists of:



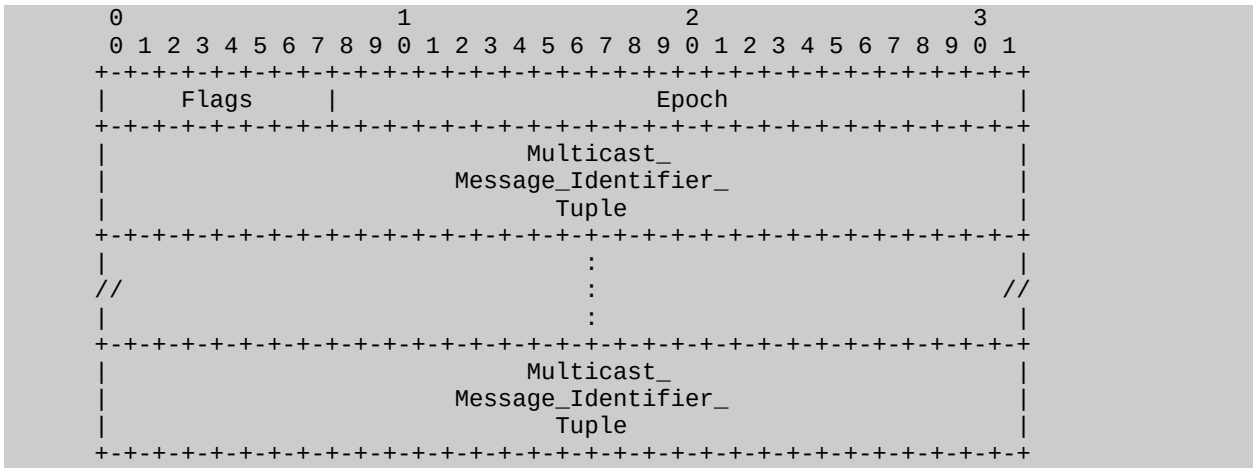
B.3.22.3 *IPv6/MESSAGE_ID SRC_LIST object: Class = 25, C_Type = 3*



Where a IPv6 Source_Message_Identifier_Tuple consists of:

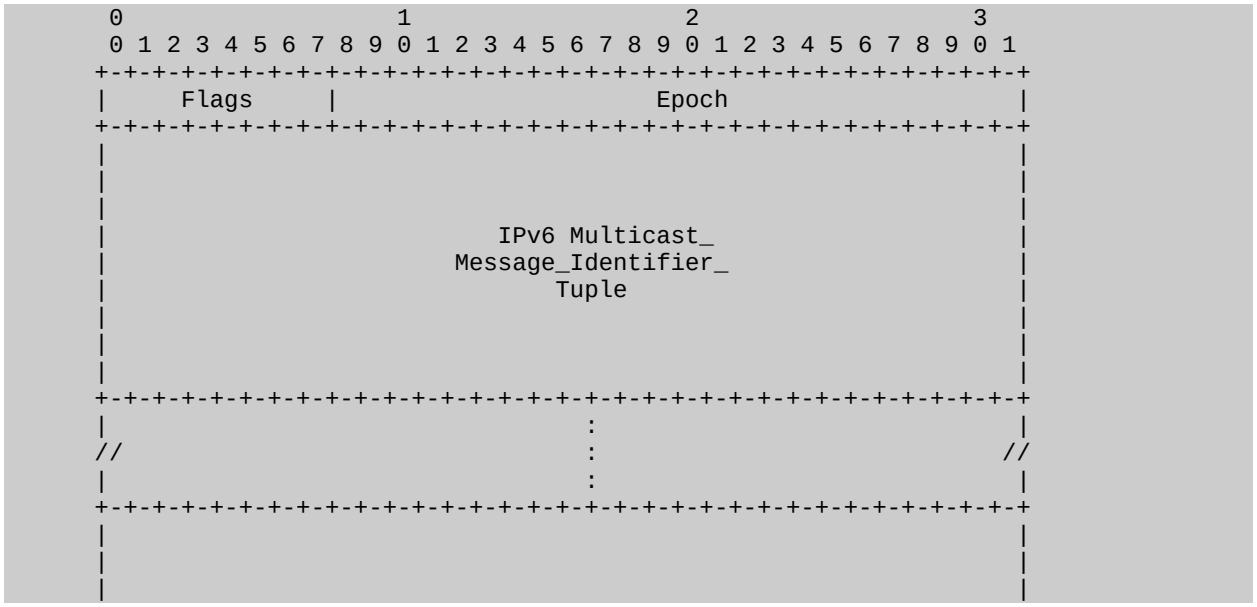


B.3.22.4 *IPv4/MESSAGE_ID MCAST_LIST object: Class = 25, C_Type = 4*



Where a Multicast_Message_Identifier_Tuple consists of:

B.3.22.5 *IPv6/MESSAGE_ID MCAST_LIST object: Class = 25, C_Type = 5*



```
|                          IPv6 Multicast_                      |
|                       Message_Identifier_                     |
|                             Tuple                             |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where a IPv6 Multicast_Message_Identifier_Tuple consists of:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Message_Identifier                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      IPv6 Source_IP_Address                   |
|                           (16 Bytes)                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    IPv6 Destination_IP_Address                |
|                           (16 Bytes)                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

B.3.23 RECOVERY LABEL Class (C-num = 34)

The Recovery_Label object is used during the nodal fault recovery process. The format of a Recovery_Label object is identical to a generalized label. A Recovery_Label object uses Class-Number 34 (of form 0bbbbbbb) and the C-Type of the label being suggested.

B.3.24 LABEL SET Class (C-num = 36)

B.3.24.1 *LABEL SET object: Class = 36 (of form 0bbbbbbb), C-Type = 1*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             | Class-Num (36)|   C-Type (1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Action     |      Reserved     |        Label Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Subchannel 1                         |
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                             :                                 :
:                             :                                 :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Subchannel N                         |
|                             ...                               |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Label Type: 14 bits
  - Indicates the type and format of the labels carried in the object. Values match the C-Type of the appropriate RSVP_LABEL object. Only the low order 8 bits are used in this field.

B.3.25 PROTECTION Class (C-num = 37)

B.3.25.1 *PROTECTION object: Class = 37, C_Type = 1*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             | Class-Num (37)|   C-Type (1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|                  Reserved                       | Link Flags|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Link Flags: 6 bits
  - Indicates desired link protection type. As previously mentioned, protection capabilities of a link may be advertised in routing. A value of 0 implies that any, including no, link protection may be used. More than one bit may be set to indicate when multiple protection types are acceptable. When multiple bits are set and multiple protection types are available, the choice of protection type is a local (policy) decision. The following flags are defined:

    0x20 Enhanced

    Indicates that a protection scheme that is more reliable than Dedicated 1+1 should be used, e.g., 4 fiber BLSR/MS-SPRING.

    0x10 Dedicated 1+1

    Indicates that a dedicated link layer protection scheme, i.e., 1+1 protection, should be used to support the LSP.

    0x08 Dedicated 1:1

    Indicates that a dedicated link layer protection scheme, i.e., 1:1 protection, should be used to support the LSP.

    0x04 Shared

    Indicates that a shared link layer protection scheme, such as 1:N protection, should be used to support the LSP.

    0x02 Unprotected

    Indicates that the LSP should not use any link layer protection.

    0x01 Extra Traffic

    Indicates that the LSP should use links that are protecting other (primary) traffic. Such LSPs may be preempted when the links carrying the (primary) traffic being protected fail.

B.3.26 SUGGESTED LABEL class (C-num = 129)

The format of a Suggested_Label object is identical to a generalized label. It is used in Path messages. A Suggested_Label object uses Class-Number 129 (of form 10bbbbbb) and the C-Type of the label being suggested.

B.3.27 ACCEPTABLE LABEL SET class (C-num = 130)

Acceptable_Label_Set objects use a Class-Number 130 (of form 10bbbbbb). The remaining contents of the object, including C-Type, have the identical format as the Label_Set object. Acceptable_Label_Set objects may be carried in PathErr and ResvErr messages. The procedures for defining an Acceptable Label Set follow the procedures for defining a Label Set. Specifically, an Acceptable Label Set is defined via one or more Acceptable_Label_Set objects. Specific labels/subchannels can be added to or excluded from an Acceptable Label Set via Action zero (0) and one (1) objects respectively. Ranges of labels/subchannels can be added to or excluded from an Acceptable Label Set via Action two (2) and three (3) objects respectively. When the Acceptable_Label_Set objects only list labels/subchannels to exclude, this implies that all other labels are acceptable.

The inclusion of Acceptable_Label_Set objects is optional. If included, the PathErr or ResvErr message SHOULD contain a "Routing problem/Unacceptable label value" indication. The absence of Acceptable_Label_Set objects does not have any specific meaning.

B.3.28 LSP_TUNNEL_INTERFACE_ID Class (C-num = 193)

B.3.28.1 *LSP_TUNNEL_INTERFACE_ID Object: Class = 193, C-Type = 1*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       LSR's Router ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface ID (32 bits)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This object can optionally appear in either a Path message or a Resv message. In the former case, we call it the "Forward Interface ID" for that LSP; in the latter case, we call it the "Reverse Interface ID" for the LSP.

B.3.29 NOTIFY REQUEST Class (C-num = 195)

Notifications may be sent via the Notify message defined below. The Notify Request object is used to request the generation of notifications. Notifications, i.e., the sending of a Notify message, may be requested in both the upstream and downstream directions. The Notify Request Object may be carried in Path or Resv Messages. The Notify_Request Class-Number is 195 (of form 11bbbbbb).

If a message contains multiple Notify_Request objects, only the first object is meaningful. Subsequent Notify_Request objects MAY be ignored and SHOULD NOT be propagated.

B.3.29.1 *IPV4 NOTIFY REQUEST object: Class = 195, C-Type = 1*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             | Class-Num (1) |  C-Type (1)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv4 Notify Node Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- IPv4 Notify Node Address: 32 bits
  - The IP address of the node that should be notified when generating an error message.

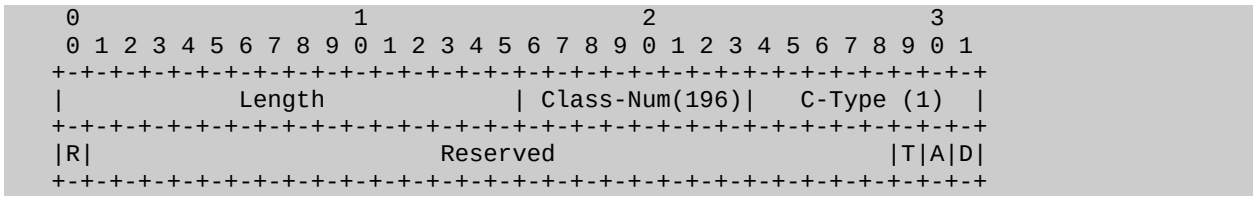
B.3.29.2 *IPV6 NOTIFY REQUEST object: Class = 195, C-Type = 2*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             | Class-Num (2) |  C-Type (2)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   IPv6 Notify Node Address                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- IPv6 Notify Node Address: 16 bytes
  - The IP address of the node that should be notified when generating an error message.

B.3.30 ADMIN_STATUS Class (C-num = 196)

B.3.30.1 *ADMIN_STATUS object: Class = 196, C_Type = 1*

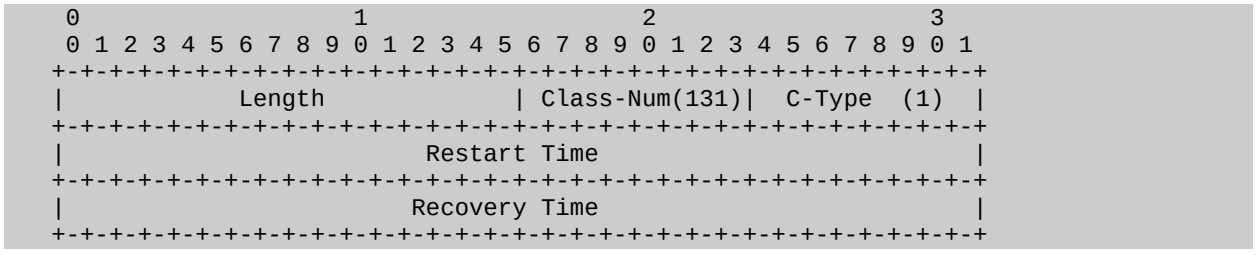


- Reflect (R): 1 bit
  - When set, indicates that the edge node SHOULD reflect the object/TLV back in the appropriate message. This bit MUST NOT be set in state change request, i.e., Notify, messages.
- Reserved: 28 bits
  - This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt. These bits SHOULD be pass through unmodified by transit nodes.
- Testing (T): 1 bit
  - When set, indicates that the local actions related to the "testing" mode should be taken.
- Administratively down (A): 1 bit
  - When set, indicates that the local actions related to the "administratively down" state should be taken.
- Deletion in progress (D): 1 bit
  - When set, indicates that that the local actions related to LSP teardown should be taken. Edge nodes may use this flag to control connection teardown.

B.3.31 RESTART_CAP Class (C-num = 131)

B.3.31.1 *RESTART_CAP object: Class = 131, C_Type = 1*

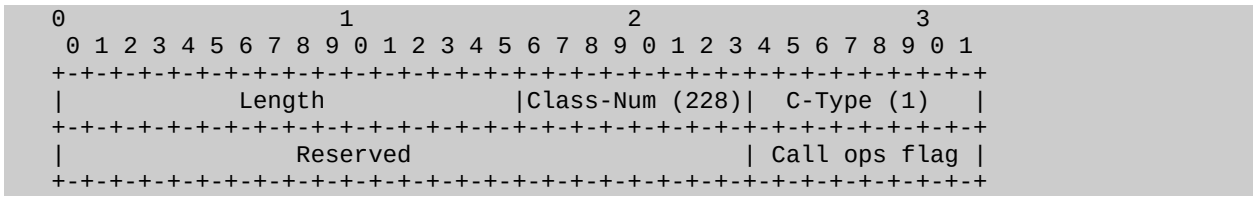
The Restart_Cap Object is carried in Hello messages.



- Restart Time: 32 bits
  - Restart Time is measured in milliseconds. Restart Time SHOULD be set to the sum of the time it takes the sender of the object to restart its RSVP-TE component (to the point where it can exchange RSVP Hello with its neighbors) and the communication channel that is used for RSVP communication. A value of 0xffffffff indicates that the restart of the sender's control plane may occur over an indeterminate interval and that the operation of its data plane is unaffected by control plane failures.
- Recovery Time: 32 bits
  - The period of time, in milliseconds, that the sender desires for the recipient to re-synchronize RSVP and MPLS forwarding state with the sender after the re-establishment of Hello synchronization. A value of zero (0) indicates that MPLS forwarding state was not preserved across a particular reboot.

B.3.32 CALL_OPS Class (C-num = 228)

B.3.32.1 *CALL_OPS object: Class = 228, C-type = 1*

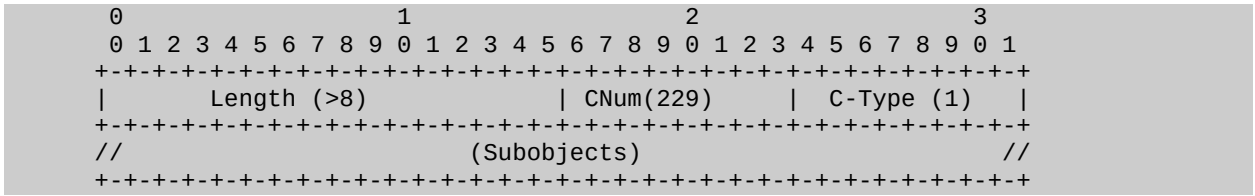


Two flags are currently defined for the "call ops flag":

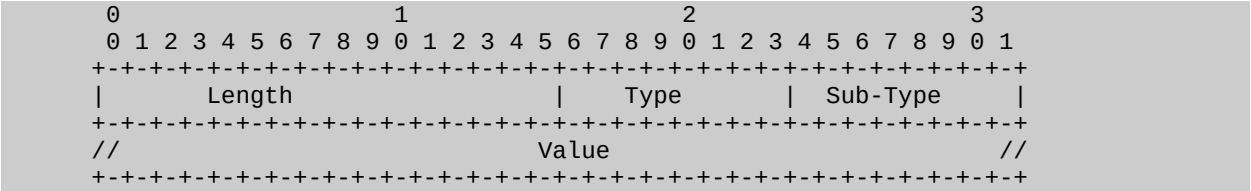
- 0x01: call without connection
- 0x02: synchronizing a call (for restart mechanism)

B.3.33 GENERALIZED_UNI Class (C-num = 229)

B.3.33.1 *Generalized_UNI object: Class = 229, C-type = 1*



The contents of a GENERALIZED_UNI object are a series of variable-length data items. The common format of the sub-objects is shown below:



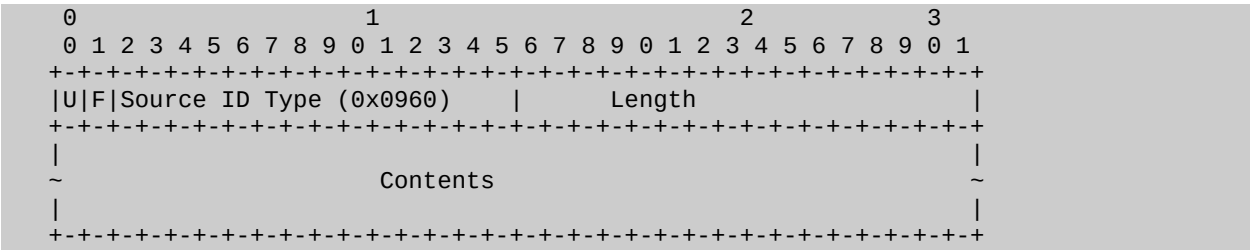
The following sub-objects are defined.

Type 1: Source Transport Network Assigned (TNA) Address sub-object

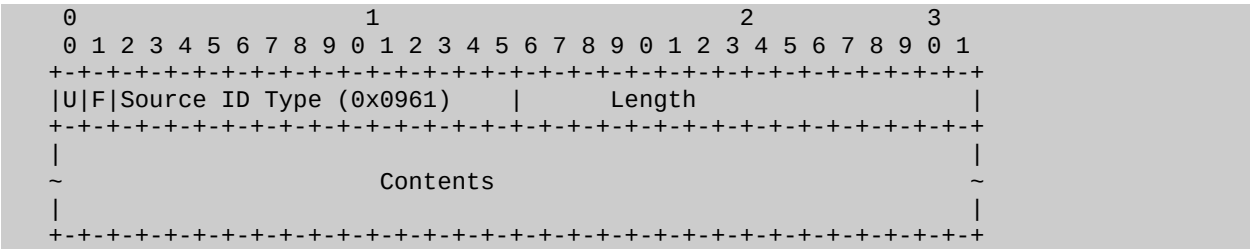
The following sub-types are defined:

- Ipv4 (Sub-type = 1);
- Ipv6 (Sub-type = 2);
- NSAP (Sub-type = 3).

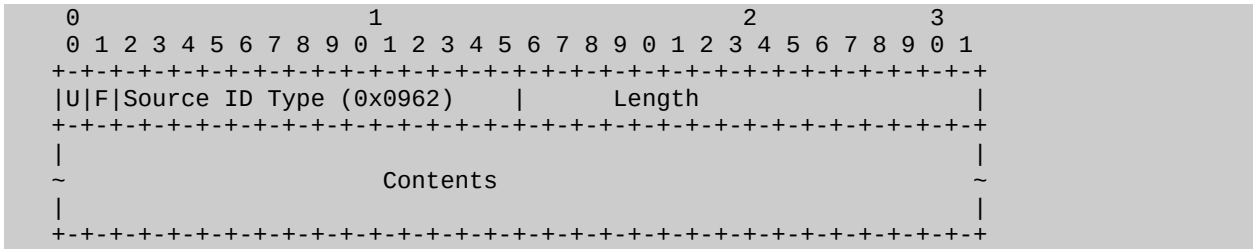
IPv4 Source ID



IPv6 Source ID



NSAP Source ID

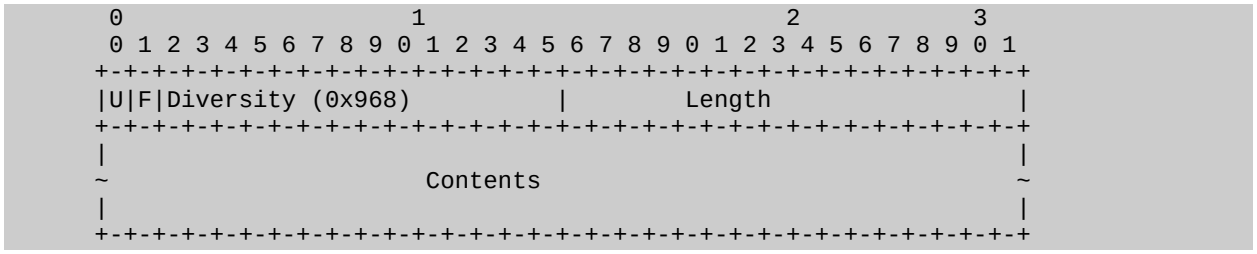


Type 2: Destination Transport Network Assigned (TNA) Address sub-object

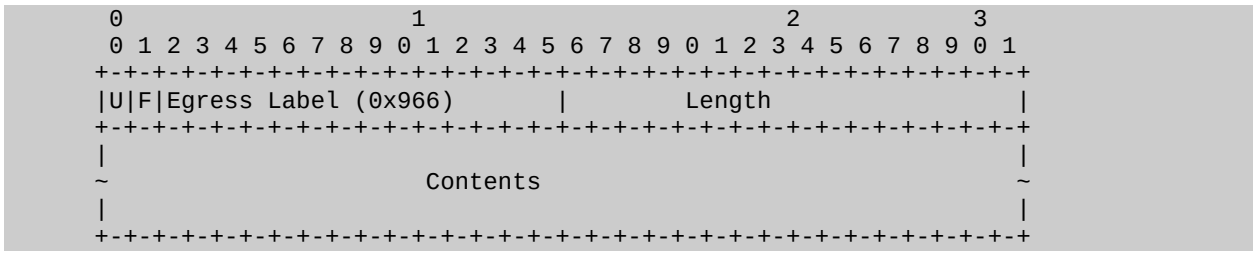
Contents are similar to those of source TNA object. The following sub-types are defined:

- Ipv4 (Sub-type = 1);
- Ipv6 (Sub-type = 2);
- NSAP (Sub-type = 3).

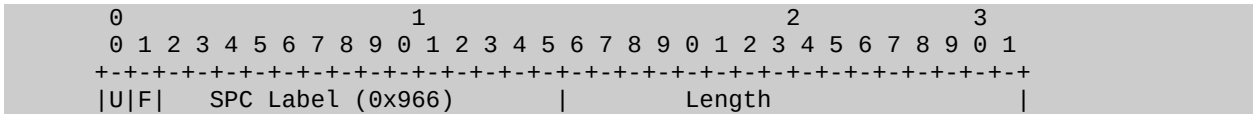
Type 3: Diversity sub-object *(Sub-type = 1)*

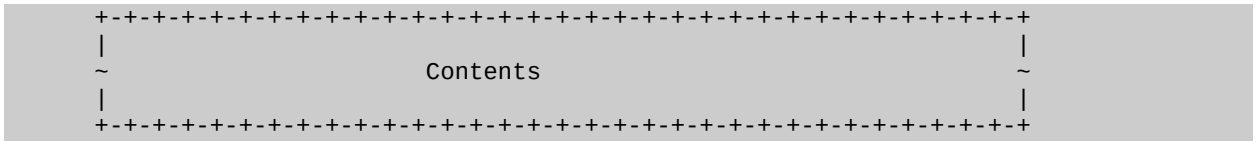


Type 4: Egress label sub-object *(Sub-type = 1)*

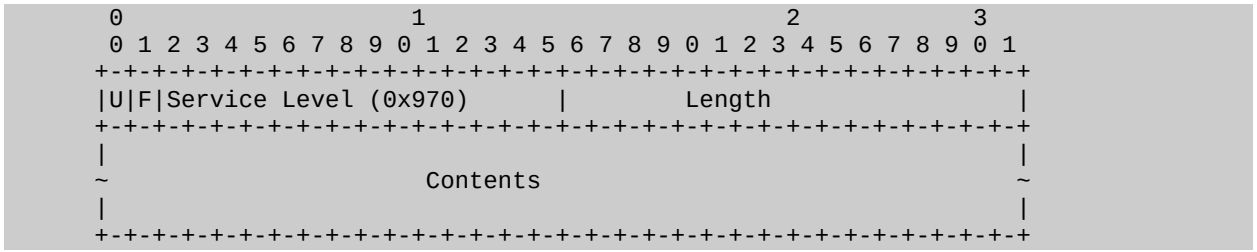


Type 4: SPC label sub-object *(Sub-type = 2)*

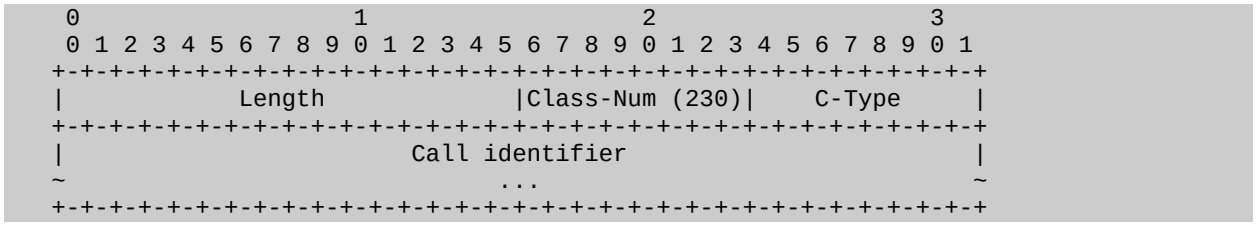




Type 5: Service level sub-object *(Sub-type = 1)*



B.3.34 CALL_ID Class (C-num = 230)



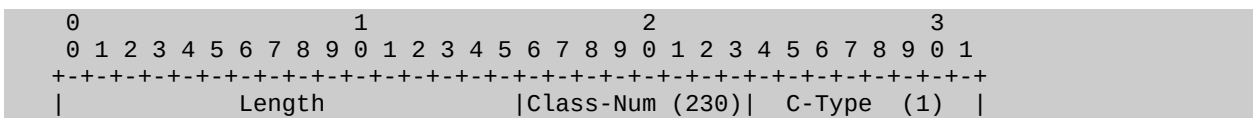
Where the following C-types are defined:

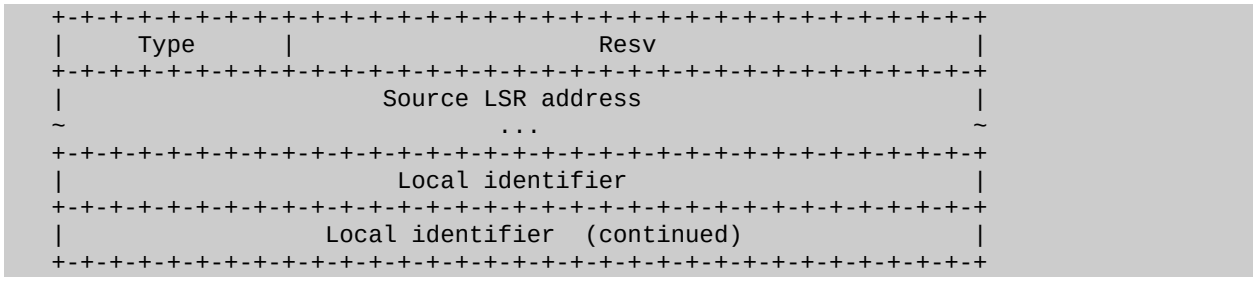
- C-Type = 1 (operator specific): The call identifier contains an operator specific identifier.
- C-Type = 2 (globally unique): The call identifier contains a globally unique part plus an operator specific identifier.

The following structures are defined for the call identifier:

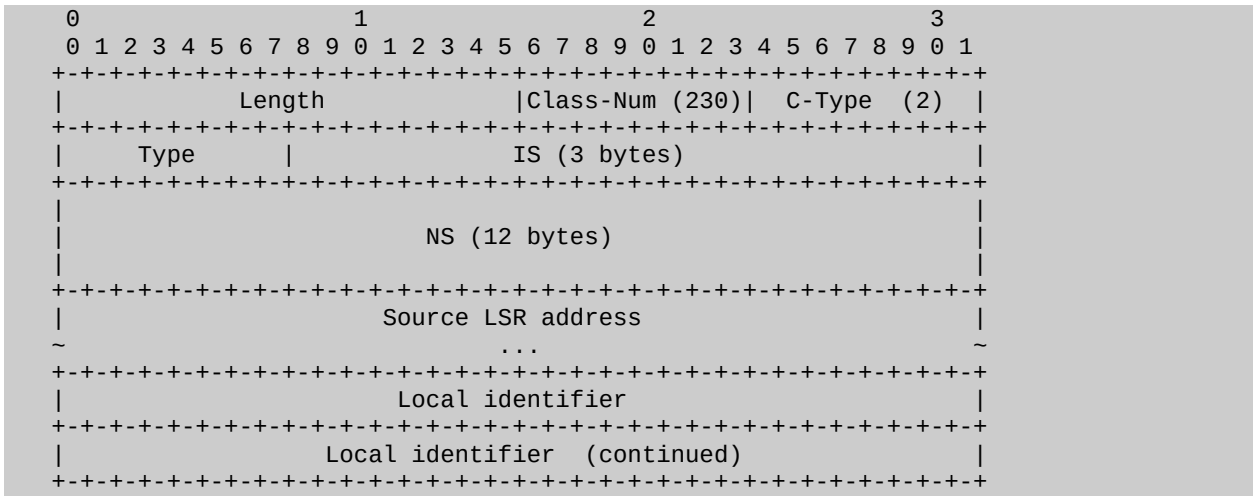
- Call identifier: generic [Length*8-32]-bit identifier.
  - The number of bits for a call identifier must be multiples of 32 bits, with a minimum size of 32 bits.

B.3.34.1 *CALL_ID Operator Specific object: Class= 230, C-Type = 1*



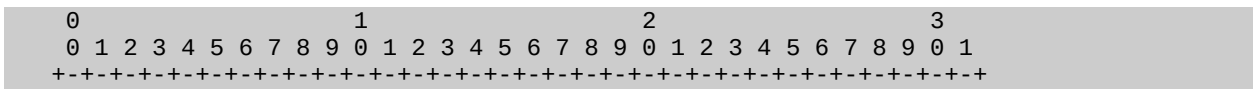


B.3.34.2 *CALL_ID Globally Unique object: Class= 230, C-Type = 2*



- Type field is defined to indicate the type of format used for the source LSR address. The Type field has the following meaning:
  For Type=0x01, the source LSR address is 4 bytes
  For Type=0x02, the source LSR address is 16 bytes
  For Type=0x03, the source LSR address is 20 bytes
  For type=0x04, the source LSR address is 6 bytes
  For type=0x7f, the source LSR address has the length defined by the vendor
- Source LSR address:
  - An address of the LSR controlled by the source network.
- Local identifier:
  - A 64-bit identifier that remains constant over the life of the call.

Format of the National Segment



```
|                       ITU carrier code                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ITU carrie dode (cont)        |  Unique access point code     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Unique access point code (continued)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

B.4 RSVP Error codes

Error Code	Name	Error Value	Name
0	Confirmation		
1	Admission Control Failure	1	Delay bound cannot be met
		2	Requested bandwidth unavailable
		3	MTU in flowspec larger than interface MTU
		4	LSP Admission Failure
		5	Bad Association Type
2	Policy Control Failure	0	Information reporting
		1	Warning
		2	Reason unknown
		3	Generic Policy Rejection
		4	Quota or Accounting violation
		5	Flow was pre-empted
		6	Previously installed policy expired (not refreshed)
		7	Previous policy data was replaced & caused rejection
		8	Policies could not be merged (multicast)
		9	PDP down or non functioning
		10	Third Party Server (e.g., Kerberos unavailable)
		11	POLICY_DATA object has bad syntax
		12	POLICY_DATA object failed Integrity Check
		13	POLICY_ELEMENT object has bad syntax
		14	Mandatory PE Missing (Empty PE is in the PD object)
		15	PEP Out of resources to handle policies
		16	PDP encountered bad RSVP objects or syntax
		17	Service type was rejected
		18	Reservation Style was rejected
		19	FlowSpec was rejected (too large
		20	Hard Pre-empted
		100	Unauthorized sender
		101	Unauthorized receiver
		102	ERR_PARTIAL_PREEMPT
		103	Inter-domain policy failure
		104	Inter-domain explicit route rejected
3	No path information for this Resv message		
4	No sender information for this Resv message		
5	Conflicting reservation style		
6	Unknown reservation style		
7	Conflicting dest ports		
8	Conflicting sender ports		
9 – 12	reserved		
12	Service preempted		

Error Code	Name	Error Value	Name
13	Unknown object class		
14	Unknown object C-Type		
15 – 19	reserved		
20	Reserved for API		
21	Traffic Control Error	1	Service conflict
		2	Service unsupported
		3	Bad Flowspec value
		4	Bad Tspec value
		5	Bad Adspec value
22	Traffic Control System error		
16	RSVP System error		
24	Routing Problem	1	Bad EXPLICIT_ROUTE object
		2	Bad strict node
		3	Bad loose node
		4	Bad initial subobject
		5	No route available toward destination
		6	Unacceptable label value
		7	RRO indicated routing loops
		8	MPLS being negotiated, but a non-RSVP-capable router stands in the path
		9	MPLS label allocation failure
		10	Unsupported L3PID
		11	Label Set
		12	Switching Type
		13	Unassigned
		14	Unsupported Encoding
		15	Unsupported Link Protection
		16	Unknown Interface Index
		17	Unsupported LSP Protection
		18	PROTECTION object not applicable
		19	Bad PRIMARY_PATH_ROUTE object
		20	PRIMARY_PATH_ROUTE object not applicable
		21	LSP Segment Protection Failed
		22	Re-routing limit exceeded
		23	Unable to Branch
		24	Unsupported LSP Integrity
		25	P2MP Re-Merge Detected
		26	P2MP Re-Merge Parameter Mismatch
		27	ERO Resulted in Re-Merge
		28	Contiguous LSP type not supported
		29	ERO conflicts with inter-domain signaling method
		30	Stitching unsupported
		31 – 63	Unassigned
		64	Unsupported Exclude Route Subobject Type
		65	Inconsistent Subobject
		66	Local Node in Exclude Route
		67	Route Blocked by Exclude Route
		68	XRO Too Complex
		69	EXRS Too Complex

Error Code	Name	Error Value	Name
		70 – 99	Unassigned
		100	Diversity not available
		101	Service level not available
		102	Invalid/Unknown connection ID
		103	No route available toward source (ASON)
		104	Unacceptable interface ID (ASON)
		105	Invalid/unknown call ID (ASON)
		106	Invalid SPC interface ID/label (ASON)
25	Notify Error	1	RRO too large for MTU
		2	RRO Notification
		3	Tunnel locally repaired
		4	Control Channel Active State
		5	Control Channel Degraded State
		6	Preferable path exists
		7	Local link maintenance required
		8	Local node maintenance required
		9	LSP Failure
		10	LSP Recovered
		11	LSP Locally Failed
26	NEW-AGGREGATE-NEEDED		
27	Diffserv Error	1	Unexpected DIFFSERV object
		2	Unsupported PHB
		3	Invalid EXP ←→ PHB mapping
		4	Unsupported PSC
		5	Per-LSP context allocation failure
28	Diff-Serv-aware TE Error	1	Unexpected CLASSTYPE object
		2	Unsupported Class-Type
		3	Invalid Class-Type value
		4	Class-Type and setup priority do not form a configured TE-Class
		5	Class-Type and holding priority do not form a configured TE-Class
		6	Class-Type and setup priority do not form a configured TE-Class AND Class-Type and holding priority do not form a configured TE-Class
		7	Inconsistency between signaled PSC and signaled Class-Type
		8	Inconsistency between signaled PHBs and signaled Class-Type
29	Unknown Attributes TLV		
30	Unknown Attributes Bit		
31	Alarms		
32	Call Management	1	Call ID Contention
		2	Connections still Exist
		3	Unknown Call ID
		4	Duplicate Call

<END-OF-DOCUMENT>