MESCAL

Management of End-to-end Quality of Service Across the Internet at Large

IST-2001-37961

D1.2: Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements

Document Identifier:	MESCAL/WP1/UniS/D1.2/Public-Final 1.0

Deliverable Type:	Report	**Contractual Date:**	30 November 2003
Deliverable Nature:	Public	**Actual Date:**	30 January 2004

Editor:	Michael Howarth, University of Surrey
Authors:	*FTR&D:* P. Morand, M. Boucadair, P. Levis *TRT:* R. Egan, H. Asgari *UCL:* D. Griffin, J. Griem, J. Spencer *UniS:* P. Trimintzios, M. Howarth, N. Wang, P. Flegkas, K.-H. Ho, S. Georgoulas, G. Pavlou *Algo:* P. Georgatsos, T. Damilatis

Abstract: This document is a key deliverable of the MESCAL project, and is the result of activity AC1.2. It specifies algorithms and protocols that enable inter-domain Quality of Service (QoS) across the Internet, including in particular:

- Algorithms and protocols that enable SLS establishment between peers and invocation of service instances across domains;
- Offline inter-domain and intra-domain traffic engineering algorithms;
- QoS enhancements to BGP for dynamic inter-domain traffic engineering;
- Integration of inter-/intra domain SLS management with traffic engineering;
- Multicast SLSs and traffic engineering algorithms.

Keywords: Inter-domain, QoS, QoS classes, SLS, traffic engineering, BGP, PCS, admission control

Project funded by the European Community under the "Information Society Technology" Programme (1998-2002)

Executive Summary

This document is a key deliverable of the MESCAL project, and is the result of activity AC1.2. The overall objective of MESCAL is to propose and validate scalable, incremental solutions that enable the flexible deployment and delivery of inter-domain Quality of Service (QoS) across the Internet. The project will validate its results through prototypes, and evaluate the overall performance through simulations and prototype testing. This document specifies algorithms and protocols that enable inter-domain Quality of Service (QoS) across the Internet, including in particular:

- Algorithms and protocols that enable Service Level Specification (SLS) establishment between peers, including domain advertisement of QoS capabilities and QoS capability discovery;
- Algorithms and protocols for invocation of service instances across domains;
- Offline inter-domain and intra-domain traffic engineering algorithms. Two inter-domain provisioning cycles are described: a longer-timescale cycle in which pSLS requirements are determined by the traffic engineering algorithms and then negotiated with peer domains, and a shorter-term cycle in which inter-domain bandwidth is invoked within the framework of existing pSLSs;
- Dynamic inter-domain traffic engineering algorithms and protocols, including QoS enhancements to BGP and a Path Computation Server (PCS) Communications Protocol (PCP) for MPLS traffic engineering;
- Algorithms for inter-domain traffic forecast;
- Algorithms that integrate inter- and intra- domain SLS management with traffic engineering, defining the data that needs to be passed between the SLS handling functional blocks, traffic forecast, and traffic engineering components;
- Algorithms for online SLS invocation handling (i.e. inter-domain admission control);
- Multicast SLS definitions and multicast traffic engineering algorithms.

Note that the algorithms and protocols specified to date are currently being validated through implementation and experimentation. For this reason some of the detailed specifications have been suppressed in the public version of this document and those that are included should be considered as drafts. D1.3 will contain the final version of all algorithms and protocol specifications.

In addition to the algorithms and protocols outlined above, this document provides an update to the MESCAL functional architecture and the business model originally defined in [D1.1]. This update describes refinements that reflect an increased depth of understanding of the issues of Inter-domain QoS delivery gained since first defining them in [D1.1]. For completeness the document also briefly summarises issues of interoperability, bidirectionality and charging that are dealt with in more detail in [D1.4].

In describing the algorithms and protocols, the structure of the document reflects the highest-level view of the MESCAL functional architecture. Each functional block of the modified functional architecture is decomposed, and its interfaces and behavioural specification described. Outline test requirements are given for the algorithms and protocols.

Table of Contents

EXECUTIVE SUMMARY **2**

1 INTRODUCTION **11**
1.1 Background 11
1.2 Role of WP1 and this deliverable 11
1.3 Structure of this document 12

2 MESCAL MODEL AND FUNCTIONAL ARCHITECTURE UPDATE **13**
2.1 Overview 13
2.2 Business model update 13
2.3 Meta-QoS classes 18
2.4 Review of the MESCAL Functional Architecture 24
2.5 Interoperability of MESCAL service options 35
2.6 Bidirectionality 40
2.7 Network Provisioning Cycle 43
2.8 MESCAL multicast functional architecture 48

3 SERVICE PLANNING AND QOS CAPABILITIES EXCHANGE **50**
3.1 Introduction 50
3.2 QoS-based Service Planning 51
3.3 QoS Capabilities Discovery and Advertisement 53

4 SLS MANAGEMENT **55**
4.1 Introduction 55
4.2 cSLS and pSLS specifications 56
4.3 Service Negotiation protocol 70
4.4 SLS Order Handling 84
4.5 pSLS Ordering 89
4.6 cSLS Ordering 93
4.7 pSLS Invocation 95
4.8 SLS Invocation Handling 97

5 TRAFFIC ENGINEERING **108**
5.1 Inter-domain TE terminology 108
5.2 Traffic Forecast 111
5.3 Traffic Engineering Interactions 116
5.4 Offline Inter-domain TE 126
5.5 Dynamic Inter-domain TE 135
5.6 IP-based Intra-domain TE 173

6 TRAFFIC ENFORCEMENT **187**
6.1 Traffic Conditioning & QC Enforcement 187
6.2 PHB Enforcement 189
6.3 IP Forwarding 190
6.4 MPLS forwarding 191

7 MULTICAST **192**
7.1 Multicast cSLS/pSLS 192
7.2 Offline Multicast Traffic Engineering (OMTE) 193
7.3 Dynamic Group Management (DGM) 195
7.4 Dynamic Multicast Routing (DMR) 197
7.5 PHB Enforcement (PE) 200
7.6 Multicast Forwarding (MF) 202
7.7 RPF Checking (RC) 202
7.8 Overall interface relationship 202

8 REFERENCES **203**

Detailed Table of Contents

EXECUTIVE SUMMARY ... 2

1 INTRODUCTION ... 11
1.1 Background ... 11
1.2 Role of WP1 and this deliverable ... 11
1.3 Structure of this document ... 12

2 MESCAL MODEL AND FUNCTIONAL ARCHITECTURE UPDATE ... 13
2.1 Overview ... 13
2.2 Business model update ... 13
2.2.1 Introduction ... 13
2.2.2 Customers and users ... 14
2.2.3 IP network providers ... 14
2.2.4 Service providers ... 15
2.2.5 Physical connectivity providers ... 16
2.2.6 Resellers ... 17
2.3 Meta-QoS classes ... 18
2.3.1 Introduction ... 18
2.3.2 Current inter-domain QoS deployment assessment ... 18
2.3.3 Requirements ... 18
2.3.4 A basic QoS inter-domain problem: binding l-QC ... 18
2.3.4.1 Problem statement ... 18
2.3.4.2 Who is a given Service Provider going to trust? ... 19
2.3.4.3 Using only local information to bind l-QCs ... 19
2.3.4.4 What will ensure the AS path consistency? ... 19
2.3.5 The Meta-QoS-Class concept ... 19
2.3.5.1 Meta-QoS-Class based on a worldwide common understanding of application QoS needs ... 19
2.3.5.2 Meta-QoS-Class definition ... 20
2.3.5.3 What's in and out of a Meta-QoS-Class? ... 20
2.3.6 The fundamental use case: the QoS Internet as a set of Meta-QoS-Class planes ... 21
2.3.7 Proposal for a first set of meta-QoS-classes ... 21
2.3.7.1 Premium Meta-QoS-Class ... 22
2.3.7.2 Gold Meta-QoS-Class (two classes) ... 22
2.3.7.3 Best Effort Meta-QoS-Class ... 22
2.3.7.4 Lower than Best Effort Meta-QoS-Class ... 22
2.3.8 Future work ... 22
2.3.8.1 Thorough definition ... 22
2.3.8.2 Standardising Meta-QoS-Classes? ... 23
2.3.8.3 Validating a set of relevant Meta-QoS-Classes ... 23
2.3.8.4 Meta-QoS-Class outclassing procedures ... 23
2.3.8.5 Security ... 23
2.3.8.6 Finding out other use cases ... 23
2.3.9 Conclusion ... 23
2.4 Review of the MESCAL Functional Architecture ... 24
2.4.1 Functional architecture overview ... 24
2.4.2 QoS-based Service Planning, QoS Capabilities Discovery and Advertisement ... 25
2.4.3 Off-line Traffic Engineering ... 25
2.4.4 Dynamic Traffic Engineering ... 27
2.4.5 SLS Management ... 27
2.4.5.1 Monitoring and SLA Assurance ... 28
2.4.5.2 Traffic Conditioning and QC Enforcement, PHB Enforcement and IP Forwarding ... 28
2.4.6 Interactions between SLS Management and Dynamic Inter-domain Traffic Engineering ... 29
2.4.6.2 Interactions ... 29
2.4.7 Other functions and capabilities ... 33
2.4.8 Updated Functional Architecture ... 33
2.5 Interoperability of MESCAL service options ... 35
2.5.1 Introduction ... 35
2.5.2 Service considerations ... 35
2.5.3 Co-existence scenario ... 35

2.5.4 Inter-working scenario ... 38
2.6 Bidirectionality ... 40
2.6.1 Bi-directionality in Statistical Guarantees Solution Option (2) ... 40
2.6.1.1 Method 1: Bi-directional pSLSs ... 40
2.6.1.2 Method 2: Multiple unidirectional cascades ... 41
2.6.2 Bi-directionality in Loose Guarantees Solution Option (1) ... 41
2.6.3 Bi-directionality in Hard Guarantees Solution Option (3) ... 41
2.6.4 Conclusion ... 42
2.7 Network Provisioning Cycle ... 43
2.7.1 Network Planning and Provisioning ... 43
2.7.2 Optical network technologies for dynamic network provisioning ... 43
2.7.3 Network Provisioning in the MESCAL functional architecture ... 44
2.7.4 Network Planning interactions ... 45
2.7.4.1 Creation of new service peers ... 45
2.7.4.2 Resource and Network Provisioning Cycles ... 46
2.7.5 Relationships between Network Planning and Traffic Engineering Algorithms ... 47
2.8 MESCAL multicast functional architecture ... 48
2.8.1 Overview ... 48

3 SERVICE PLANNING AND QOS CAPABILITIES EXCHANGE ... 50
3.1 Introduction ... 50
3.2 QoS-based Service Planning ... 51
3.2.1 Objectives ... 51
3.2.2 Interface Specification ... 51
3.2.3 Behaviour Specification ... 52
3.3 QoS Capabilities Discovery and Advertisement ... 53
3.3.1 Objectives ... 53
3.3.2 Interface Specification ... 53
3.3.2.1 External Interface ... 53
3.3.2.2 Internal Interface ... 54
3.3.3 Behaviour Specification ... 54

4 SLS MANAGEMENT ... 55
4.1 Introduction ... 55
4.2 cSLS and pSLS specifications ... 56
4.2.1 Introduction ... 56
4.2.2 Types of pSLS and Specification Requirements ... 56
4.2.3 A general model for pSLS and QoS-based services ... 58
4.2.3.1 SLS-T Specifications ... 59
4.2.3.2 SSS-T Specifications ... 63
4.2.4 MESCAL pSLS models ... 65
4.3 Service Negotiation protocol ... 70
4.3.1 Introduction ... 70
4.3.2 Negotiation Protocol Requirements ... 71
4.3.3 Negotiation Model ... 71
4.3.4 SrNP Overview ... 72
4.3.5 SrNP Messages and Interface ... 73
4.3.5.1 Protocol Messages ... 73
4.3.5.2 Message Sequence Charts (MSCs) ... 76
4.3.5.3 Interface (I/F) Messages ... 76
4.3.6 SrNP Finite State Machine (FSM) ... 77
4.3.6.1 Timers ... 77
4.3.6.2 Events ... 77
4.3.6.3 SrNP Client FSM ... 78
4.3.6.4 SrNP Server FSM ... 80
4.3.7 Implementation Aspects ... 81
4.3.7.1 Protocol Stack ... 81
4.3.7.2 Protocol Interface ... 82
4.3.8 Scalability and Stability Analysis ... 82
4.3.9 Conclusions ... 83
4.4 SLS Order Handling ... 84
4.4.1 Objectives ... 84
4.4.2 Interface Specification ... 84

4.4.2.1 External Interface 84
4.4.2.2 Internal Interfaces....85
4.4.3 Behaviour Specification 85
4.4.3.1 Subscription Admission Control Algorithm 86
4.4.4 Test requirements 88
4.5 pSLS Ordering.... 89
4.5.1 Objectives.... 89
4.5.2 Interface Specifications.... 89
4.5.2.1 External interface with the Order Handling components of peering ASs 89
4.5.2.2 Internal interface with the Binding Selection function block 90
4.5.3 Behaviour Specification 90
4.5.3.1 Transaction oriented negotiations logic....91
4.5.4 Test requirements 92
4.6 cSLS Ordering.... 93
4.6.1 Objectives.... 93
4.6.2 Interface Specification 93
4.6.2.1 External interface with the end consumers 93
4.6.2.2 Internal interface with the QoS Based Service Planning component....93
4.6.2.3 Internal interface with the Order Handling component 93
4.6.3 Behaviour Specification 94
4.6.4 Test requirements 94
4.7 pSLS Invocation 95
4.7.1 Objectives.... 95
4.7.2 Interface specification.... 95
4.7.3 Behavioural specification.... 95
4.7.4 Test Requirements.... 96
4.8 SLS Invocation Handling 97
4.8.1 Objectives.... 97
4.8.2 Interface specification.... 97
4.8.3 Behavioural specification.... 99
4.8.3.1 Inputs/Outputs 99
4.8.3.2 Process Description....100
4.8.4 SLS Invocation handling issues 101
4.8.4.1 Case Studies 101
4.8.4.2 Examples....102
4.8.5 SLS Invocation Handling Algorithms.... 102
4.8.5.1 Real-time Traffic Admission Control....102
4.8.5.2 Elastic Traffic....106
4.8.6 Test Requirements.... 107

5 TRAFFIC ENGINEERING 108
5.1 Inter-domain TE terminology 108
5.1.1 Introduction.... 108
5.1.2 Definitions.... 108
5.1.2.1 $pSLS_{in}$ and $pSLS_{out}$....108
5.1.2.2 eTM and iTM 109
5.1.2.3 eRAM, iRAM and RAM 109
5.1.2.4 Bandwidth buffer 110
5.2 Traffic Forecast.... 111
5.2.1 Objectives.... 111
5.2.2 Interface Specification 111
5.2.3 Behavioural Specification 112
5.2.3.1 Functional Decomposition....112
5.2.3.2 Demand Aggregation and Derivation....113
5.2.4 Test Requirements.... 115
5.3 Traffic Engineering Interactions 116
5.3.1 Decomposition of Offline Inter-domain Traffic Engineering.... 116
5.3.2 Resource Provisioning Cycles 116
5.3.2.1 Definitions....116
5.3.2.2 Issues 117
5.3.3 Decoupled and integrated approaches to Inter- and Intra-domain TE 118
5.3.3.1 Decoupled Inter-domain Resource Optimisation 118
5.3.3.2 Integrated Inter-domain Resource Optimisation 122

5.3.4 Off-line inter-domain TE cases ... 124
5.3.4.1 Single/Multiple egress point selection ... 124
5.3.4.2 Single/Multiple pSLS$_{out}$ selection ... 124
5.3.4.3 Single/Multiple l-QC selection ... 125
5.3.4.4 Single/Multiple intra-domain route selection ... 125
5.3.4.5 QoS parameters consideration ... 125
5.3.4.6 Traffic engineering scenarios ... 125
5.4 Offline Inter-domain TE ... 126
5.4.1 Binding Selection ... 126
5.4.1.1 Introduction ... 126
5.4.1.2 Objectives ... 126
5.4.1.3 Interface specification ... 127
5.4.2 Binding Activation ... 130
5.4.2.1 Introduction ... 130
5.4.2.2 Objectives ... 130
5.4.2.3 Interface specification ... 130
5.4.3 Inter-domain Resource Optimisation ... 132
5.4.3.1 Objectives ... 132
5.4.3.2 Interface specification ... 133
5.5 Dynamic Inter-domain TE ... 135
5.5.1 QoS-inferred BGP (q-BGP) ... 135
5.5.1.1 Introduction ... 135
5.5.1.2 Definitions ... 136
5.5.1.3 Objectives and Needs ... 136
5.5.1.4 Towards a QoS-inferred BGP ... 136
5.5.1.5 q-BGP specification ... 148
5.5.2 PCS Communication Protocol ... 156
5.5.2.1 Introduction ... 156
5.5.2.2 PCS review ... 156
5.5.2.3 PCS service ... 157
5.5.2.4 The PCS Communication Protocol (PCP) ... 160
5.5.2.5 Exchange of PCP messages ... 166
5.5.2.6 Test objectives ... 171
5.6 IP-based Intra-domain TE ... 173
5.6.1 Introduction ... 173
5.6.1.1 Overall Objectives ... 173
5.6.1.2 Decomposition of Functionality ... 174
5.6.2 Resource Optimisation ... 175
5.6.2.1 Objectives ... 175
5.6.2.2 Interface Specification ... 175
5.6.2.3 Algorithm Description ... 176
5.6.2.4 Test requirements ... 181
5.6.3 Network Reconfiguration Scheduler ... 182
5.6.3.1 Objectives ... 182
5.6.3.2 Interface Specification ... 182
5.6.3.3 Algorithm Description ... 184
5.6.3.4 Test requirements ... 186

6 TRAFFIC ENFORCEMENT ... 187
6.1 Traffic Conditioning & QC Enforcement ... 187
6.1.1 Objectives ... 187
6.1.2 Interface Specification ... 187
6.1.2.1 Traffic Conditioning & QC Enforcement Interface to SLS Invocation Handling ... 187
6.1.2.2 Traffic Conditioning & QC Enforcement Interface to Dynamic Inter-and Intra- domain TE ... 188
6.2 PHB Enforcement ... 189
6.2.1 Interface Specification ... 189
6.2.1.1 PHB Enforcement Interface to Dynamic Inter-domain Traffic Engineering ... 189
6.2.2 Behavioural Specification ... 190
6.2.2.1 Description of Functions ... 190
6.3 IP Forwarding ... 190
6.3.1 Interface Specification ... 190
6.3.1.1 IP Forwarding Interface to Dynamic Inter- and Intra- domain TE ... 191
6.4 MPLS forwarding ... 191

7 MULTICAST ... 192
7.1 Multicast cSLS/pSLS ... 192
7.1.1 Introduction ... 192
7.2 Offline Multicast Traffic Engineering (OMTE) ... 193
7.2.1 Introduction ... 193
7.2.2 Interface Specifications ... 193
7.2.3 Behavioural Specification ... 194
7.2.3.1 M-ISIS based multicast TE ... 194
7.3 Dynamic Group Management (DGM) ... 195
7.3.1 Introduction ... 195
7.3.2 Interface Specification ... 196
7.3.3 Behavioural Specification ... 196
7.4 Dynamic Multicast Routing (DMR) ... 197
7.4.1 Introduction ... 197
7.4.2 Interface Specification ... 198
7.4.3 Behavioural Specification ... 199
7.4.3.1 Intra-domain DMR ... 199
7.4.3.2 Inter-domain DMR ... 200
7.5 PHB Enforcement (PE) ... 200
7.5.1 Introduction ... 200
7.5.2 Interface Specification ... 200
7.5.3 Behavioural Specification ... 200
7.6 Multicast Forwarding (MF) ... 202
7.6.1 Introduction ... 202
7.6.2 Interface Specification ... 202
7.6.3 Behavioural Specification ... 202
7.7 RPF Checking (RC) ... 202
7.7.1 Introduction ... 202
7.8 Overall interface relationship ... 202

8 REFERENCES ... 203

List of Figures

Figure 1. The MESCAL business model from D1.1 13
Figure 2. Revised MESCAL business model – Common Physical Connectivity Provider........ 16
Figure 3. Revised MESCAL business model – Interworking between Physical Connectivity Providers........ 17
Figure 4. The initial MESCAL functional architecture........ 24
Figure 5. Decomposition of the Offline inter-domain TE........ 26
Figure 6. Two adjacent autonomous systems........ 30
Figure 7. The case of bandwidth in qBGP 31
Figure 8. Illustration of i-qBGP and e-qBGP dynamic traffic engineering 32
Figure 9. Updated functional architecture........ 33
Figure 10. End-to-end uni-directional QoS service implementation 40
Figure 11. Positioning of Network Planning in the MESCAL functional architecture 45
Figure 12. Interactions – creation of new service peers 45
Figure 13. Interactions: Intra- and Inter-domain Network Provisioning Cycles 46
Figure 14. MESCAL multicast functional architecture........ 49
Figure 15. Service Planning and QoS Capabilities Exchange 50
Figure 16. QoS-based Service Planning 51
Figure 17. Advertisement and Discovery 53
Figure 18. The MESCAL functional architecture........ 55
Figure 19. Model of pSLS for Internet access at loose QoS 67
Figure 20. Model of pSLS for loose QoS tunnels in the Internet 67
Figure 21. Model of pSLS for traffic inter-exchange at a loose QoS........ 68
Figure 22. Model of pSLS for loose QoS tunnel extension 68
Figure 23. Model of pSLS for Internet access at a statistically guaranteed QoS 69
Figure 24. Model of pSLS for statistically guaranteed QoS tunnels in the Internet........ 69
Figure 25. MSCs of successful negotiations........ 76
Figure 26. MSCs of unsuccessful negotiations........ 76
Figure 27. The SrNP client FSM state transition diagram........ 78
Figure 28. The SrNP server FSM state transition diagram 80
Figure 29. SrNP protocol stacks 81
Figure 30. SLS Order Handling 84
Figure 31. Resource availability buffer 87
Figure 32. Satisfaction Level (SL), subscription admission control policy 87
Figure 33. pSLS Ordering........ 89
Figure 34. Transaction-oriented pSLS negotiations........ 91
Figure 35. cSLS Ordering........ 93
Figure 36. pSLS Invocation........ 95
Figure 37. The SLS Invocation Handling interfaces......... 97
Figure 38. SLS Invocation Handling Inputs/Outputs......... 99
Figure 39. Traffic origination/destination cases........ 108
Figure 40. The pSLS$_{in}$ and pSLS$_{out}$ sets in a domain........ 109
Figure 41. Functional relationships of Traffic Forecast........ 111
Figure 42. Functional decomposition of Traffic Forecast 112
Figure 43. Decomposition of Offline Inter-domain TE 116
Figure 44. Resource Provisioning Cycles........ 117
Figure 45. Decoupled Inter-domain Resource Optimisation........ 119
Figure 46. Integrated Inter-domain Resource Optimisation 122
Figure 47. Binding Selection interfaces........ 127
Figure 48. Binding Activation interfaces 131
Figure 49. Inter-domain Resource Optimisation interfaces........ 133
Figure 50. Reachability information exchange a la loose solution option........ 137
Figure 51. Only meta-QoS-class identifiers are carried in the q-BGP messages......... 139
Figure 52. Use of meta-QoS-class identifier and end-to-end QoS characteristics. 140
Figure 53. Use of meta-QoS-class identifier and dynamic end-to-end QoS characteristics 141
Figure 54. The statistical solution option operational mode........ 143
Figure 55. The statistical solution option operational mode-bis........ 145
Figure 56. Interaction between off-line TE and dynamic inter-domain TE......... 145
Figure 57. PCS communication........ 146

Figure 58. Route selection process required information per group. *147*
Figure 59. Towards convergent q-BGP-bis. *148*
Figure 60. q-BGP in case of MESCAL solution options. *148*
Figure 61. QoS service capability attribute. *149*
Figure 62. QoS_NLRI attribute. *151*
Figure 63. Overview *157*
Figure 64. PCS interfaces *158*
Figure 65. PSP state diagram *172*
Figure 66. MESCAL Functional Architecture, highlighting Offline Intra-domain TE *173*
Figure 67. Decomposed Intra-domain Traffic Engineering *174*
Figure 68. Interactions of the Resource Optimisation block *175*
Figure 69. Interactions of the Network Reconfiguration Scheduler Block *182*
Figure 70. M-ISIS based multicast traffic engineering *195*
Figure 71. Per-QC tree vs. hybrid tree *198*
Figure 72. Inter-domain group address swapping *201*
Figure 73. Overall multicast interface relationship. *202*

List of Tables

Table 1. Summary of data transferred from pSLSs to qBGP *32*
Table 2. SrNP protocol messages. *74*
Table 3. Parameters of the SrNP protocol messages. *75*
Table 4. SrNP interface messages *77*
Table 5. The SrNP client FSM state transition table *79*
Table 6. Description of the SrNP client FSM actions *79*
Table 7. The SrNP server FSM state transition table *80*
Table 8. Description of the SrNP server FSM actions *81*
Table 9. Summary of the loose solution option recommendations and requirements *142*
Table 10. A very simplified q-RIB example *144*
Table 11. Compatible (code, sub-code) pairs *152*
Table 12. SSM QC encoding table *196*

1 INTRODUCTION

1.1 Background

This document has been produced as part of Work Package 1 of the EU IST MESCAL project. The overall objective of MESCAL is to propose and validate scalable, incremental solutions that enable the flexible deployment and delivery of inter-domain Quality of Service (QoS) across the Internet. MESCAL will validate its results through prototypes, and evaluate the overall performance through simulations and prototype testing.

MESCAL adopts a phased approach and the technical work is split into three work packages (WPs):

- WP1, *Specification of Functional Architecture, Algorithms and Protocols,* is responsible for defining business models and the generic, multi-domain, multi-service IP QoS functional architecture for inter-domain QoS delivery. Based on these models WP1 will develop algorithms and protocols for negotiation and establishment of inter-domain service level specifications (SLSs), and will enhance and extend inter-domain traffic engineering (TE) mechanisms and routing protocols, including the required interactions with intra-domain functionality. WP1 will also define algorithm test requirements. Based on implementation experience and experimental results fed back from WP2 and WP3, later activities within WP1 will validate the initial specifications and derive enhancements as appropriate.
- WP2, *System Design and Implementation,* is responsible for undertaking basic enhancements of experimental Linux-based routers and developing simulation tools to model the general inter-domain and QoS requirements of the project. Based on the specifications from WP1, WP2 will specify the engineering approach, conduct detailed implementation design and finally implement both testbed prototypes and simulation environments.
- WP3, *Integration, Validation and Experimentation,* is responsible for configuring the required experimentation infrastructure and for conducting validation and performance evaluation activities on the prototypes and simulators developed by WP2 according to the test requirements identified by WP1. Experimentation will be executed both in the MESCAL testbed (with the support of extended development environments at other partners' premises) and using the simulators.

1.2 Role of WP1 and this deliverable

WP1, *Specification of Functional Architecture, Algorithms and Protocols*, comprises three activities. In the first, AC1.1, *Inter-domain Business Models and System Architecture,* business models have been defined and an overall functional architecture for inter-domain QoS-based services has been developed, starting from the requirements, assumptions and state of the art in this area. The second activity, AC1.2, *Algorithm and Protocol Specification,* starts from the functional architecture produced in AC1.1 and will specify algorithms and protocols for: peer SLS establishment and invocation of service instances across domains; QoS enhancements to BGP; consideration of alternative, novel approaches (e.g. link state-based); integrated inter- and intra-domain SLS management and traffic engineering; multicast SLSs and traffic engineering; impact of IPv6 on traffic engineering possibilities; and information models, algorithms and protocols for an overall policy-driven system approach. The third activity, AC1.3, *Enhancements to Algorithms and Protocol Specifications,* will produce modifications and enhancements to the AC1.2 algorithms and specifications, based on feedback from simulation and implementation experience in WP2 and WP3.

This document is the main result of activity AC1.2, *Algorithm and Protocol Specification*. The document includes the following principal components:

- Algorithms and protocols that enable SLS establishment between peers and invocation of service instances across domains;
- Offline inter-domain and intra-domain traffic engineering algorithms;
- QoS enhancements to BGP to support dynamic inter-domain traffic engineering;

- Algorithms that integrate inter- and intra-domain SLS management with traffic engineering;
- Definitions of multicast SLSs and traffic engineering algorithms, in addition to unicast capabilities.

Note that the above algorithms and protocols are currently being validated through implementation and experimentation. For this reason some of the detailed specifications have been suppressed in the public version of this document and those that are included should be considered as drafts. D1.3 will contain the final version of all algorithms and protocol specifications.

1.3 Structure of this document

The document uses the highest-level view of the MESCAL functional architecture as defined in [D1.1] Section 6.1 as the basis for the presentation of algorithms and protocols. The rest of this document is thus structured as follows:

- Section 2, *MESCAL model and Functional Architecture update*, describes refinements that reflect an increased depth of understanding of the issues of Inter-domain QoS delivery gained since first defining them in [D1.1]. These refinements encompass the MESCAL business model and the Functional Architecture, and show how the MESCAL approach fits into the Network Provisioning Cycle of Internet Providers. The Section also extends the MESCAL Functional Architecture to include the delivery of multicast QoS, by introducing a multicast QoS model. For completeness the Section also briefly summarises issues of interoperability, bidirectionality and charging that are dealt with in more detail in [D1.4].
- Section 3, *Service Planning and QoS Capabilities Exchange*, provides a decomposition of each functional block in the Service Planning functional group. The Section describes approaches for defining service offerings (including the planning of l-QCs and e-QCs), and for providing traffic demand estimates to the Traffic Forecast functional block. The Section also describes the mechanisms by which a provider advertises QoS capabilities to peer providers and in turn discovers their capabilities; the structure of these advertisements is also defined.
- Section 4, *SLS Management*, provides a decomposition of each functional block in the SLS Management functional group. The Section begins by defining the formats of the Service Subscription Specification (SSS) and SLS for Inter-domain unicast and multicast traffic. The Section then proceeds to present algorithms and protocols for pSLS and cSLS ordering and order handling. This includes the definition of an enhanced Inter-domain Service Negotiation Protocol (SrNP), based on the Intra-domain SrNP originally defined in [TEQUI, D1.4]. The Section then presents algorithms for dynamic invocation and invocation handling (i.e. admission control) for real-time traffic.
- Section 5, *Traffic Engineering*, provides a decomposition of each functional block in the TE functional group, including the detailed interfaces with other functional blocks. The Section starts by defining new terminology relevant to Inter-domain TE, and by discussing the interactions between Inter-domain and Intra-domain TE including the relationship between resource provisioning cycles for Inter- and Intra-domain TE. The Section then proceeds to describe how traffic forecasting supports the integration of Inter-domain SLS management with inter-domain TE as a component of the resource provisioning cycle. It then proceeds to present novel offline traffic engineering algorithms (both Inter-domain and Intra-domain). The Section then presents dynamic Inter-domain TE algorithms and protocols, discussing issues in QoS enhancements to BGP (qBGP) and describing proposed modifications to the protocol. A Path Computation Server (PCS) Communication Protocol (PCP) is also proposed.
- Section 6, *Traffic Enforcement*, describes changes in the Data Plane functional blocks that are a consequence of the Inter-domain QoS functions described in Sections 3-5. The areas covered include QC enforcement (classification and traffic conditioning), IP/MPLS forwarding (principally the influence of qBGP on the forwarding information base (FIB) and its interaction with the routing information base (RIB)), and PHB enforcement.
- Section 7, *Multicast*, integrates the multicast components of the MESCAL functional architecture, describing the functions and interfaces of each functional block.

2 MESCAL MODEL AND FUNCTIONAL ARCHITECTURE UPDATE

2.1 Overview

An initial specification of the MESCAL business model and functional architecture was presented in deliverable D1.1. This was subject to refinement as the problem area and its solutions were studied in more detail during the specification phase of the algorithms and protocols for the components that constitute the overall system architecture. This Section presents the refinements that have been defined to date. A final version of the system concepts, models and architectures will be documented in the final report of WP1 - D1.3 - following the detailed design, implementation and experimental work of the project.

This Section contains subsections on: updates to the MESCAL business model; an overview of the use of Meta-QoS-Classes which form the foundation of the loose QoS guarantees in MESCAL solution option 1; a revised functional architecture which captures the current view of the components necessary to implement intra- and inter-domain QoS and the interactions required between those components; a section summarising the issues of providing bi-directional QoS in a cascaded model of interdomain services – this topic is treated in more detail in deliverable D1.4; a section on how dynamic network provisioning may be achieved through a network planning component and a Network Provisioning Cycle to augment the intra- and interdomain Resource Provisioning Cycles; and a section on the additional components and additional features of existing components of the functional architecture which are required to support multicast QoS across domains.

2.2 Business model update

2.2.1 Introduction

The business model assumed by MESCAL in deliverable D1.1 [D1.1] is illustrated in Figure 1. The model depicts from the perspectives of MESCAL the stakeholders involved in the chain of QoS-based service delivery in the Internet.

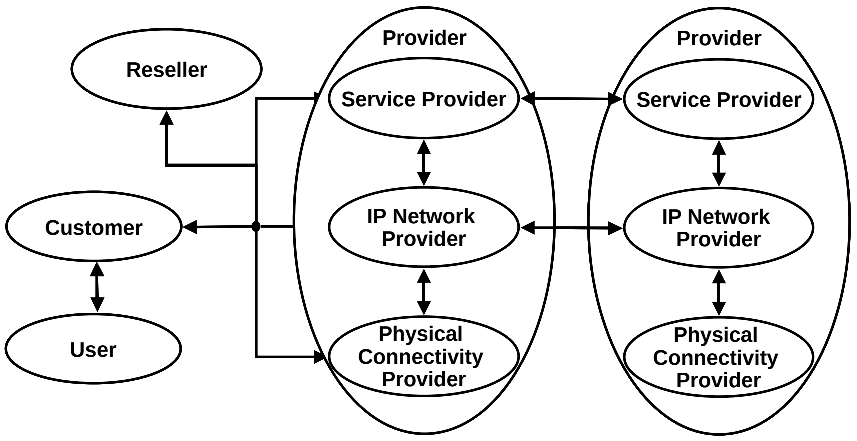


Figure 1. The MESCAL business model from D1.1

The broad classes of business relationships described by this model are those identified between the different entities involved in interdomain QoS delivery. As stated in deliverable D1.1, the focus of the MESCAL project is on the interactions between the set of IP Network Providers (ISPs) involved in the end-to-end delivery of QoS-based IP services, i.e. across multiple domains. A large number of ISPs can be involved in the provision of global IP connectivity services. The necessary business relationships and roles between the set of IP Network Providers is analysed in some detail in deliverable D1.4 [D1.4]. This is summarised in the following subsections as is the definition of the role of the other stakeholders that form the overall MESCAL business model.

2.2.2 Customers and users

A *'Customer'* (subscriber) denotes an entity, which has the legal ability to subscribe to QoS-based services offered by *'Providers'*. *'Customers'* are the target recipients of QoS-based services. They interact with *'Providers'* (or *'Resellers'*, see below) following a customer-provider paradigm, with the purpose to 'buy' services to meet their communication needs and requirements.

A *'User'* is an entity (human being or a process from a general perspective), which has been named by a *'Customer'* and appropriately identified by a *'Provider'* for actually requesting/accessing and using the QoS-based services bought by the *'Customer'*. The use of the services should be in line with the terms and conditions agreed in the SLA between the *'Customer'* and the *'Provider'*. In essence, *'Users'* are the end-users of the services and they can only exist in association with a *'Customer'*.

2.2.3 IP network providers

'IP Network Providers' offer QoS-based plain IP connectivity services, that is services, which provide reachability between hosts in the IP address space. Such *'Providers'* must own and administer an IP network infrastructure. For connecting customers to their IP infrastructure, *'IP Network Providers'* may interact with separate *'Access Providers'* – a role which isn't explicitly covered in a separate entity in the business model but may be considered to be a provider with the physical connectivity provider role only. Alternatively, customers could be connected through means/facilities provided by the *'IP Network Providers'* themselves.

'IP Network Providers' may be differentiated according to the geographical span of their IP network infrastructure. As such, we may distinguish between small, medium and large *'IP Network Providers'*, with this distinction being relative (to a given area size) rather than absolutely defined. For example, considering a continental area, small, medium, large *'IP Network Providers'* may be thought as regional (covering specific cities of a country), national (covering a specific country), continental (covering specific countries of the continent) respectively.

Based on this distinction the current business model of the best effort Internet is built around a three tier hierarchy, with the business relationships between the providers being determined by their relative position in this hierarchy [HUST, D1.4]. In order to provide access to the global Internet, *'IP Network Providers'* must interact with each other; there cannot be a single provider offering global Internet coverage. Currently, in the best-effort Internet, there exist two forms of distinct relationships between *'IP Network Providers'* for traffic exchange, underlined by respective business agreements: *peering* and *transit*. Peering is termed as the business relationship, whereby *'IP Network Providers'* reciprocally provide only access to each other's customers. Peering is a non-transitive relationship. Peering is a mutual agreement between *'IP Network Providers'* to exchange data between themselves, normally for no fee or charge. Transit is the business relationship, whereby one transit provider provides access to all destinations in its routing table (could be global Internet) to another *'IP Network Provider'* for a charge. It should be clarified that the term 'peering' is also used in this document to denote that two providers interact with each other for the purpose of expanding the topological scope of their offered services, under any business relationship which may govern this interaction; it should be not be taken that this implies a specific peering business relationship as defined above.

The MESCAL solution adopts a *hop-by-hop, cascaded model* for the interactions between NPs both at the service and network (IP) layers. Service layer interactions result in the establishment of service agreements between NPs, *pSLSs* in MESCAL terminology, aggregating customer service traffic, which need to be supported by appropriate service management and traffic engineering capabilities per provider domain as well as by BGP-based interactions at the IP layer for QoS inter-domain routing purposes.

The type of inter-domain relationships and interactions impacts the service negotiation procedures, the required signalling protocols, the QoS binding, and path selection. The following approaches are considered in detail in deliverable D1.4 [D1.4]:

- The *centralised* approach where a Network Provider negotiates *pSLSs* directly with an appropriate number of downstream providers to construct an end-to-end QoS service. With this approach, service peers are not necessarily BGP peers.
- The *cascaded* approach where a NP only negotiates *pSLS*s with its immediate neighbouring provider/s to construct an end-to-end QoS service. With this approach, service peers are also BGP peers.
- The *hub* approach, which is similar to the centralised approach, where the Service Provider (SP), as a distinct entity from NP, is the central point that negotiates and establishes *pSLSs*.
- The *hybrid* approach, which is the mixture of centralised and cascaded approaches.

Within the MESCAL project, the first two major approaches have been considered for further study in order to construct end-to-end QoS-based services across the Internet at large scale. A single point of control for the service instances is the compelling feature of the centralised approach. The use of the centralised approach for more than a few interconnected NPs would be increasingly difficult to manage. Providers would prefer to offer services which reflect current Internet structure and for whom the use of the centralised approach would be inappropriate in many instances. Such providers would probably consider using the cascaded approach, which reflects the loosely coupled structure of Internet. Within the context of MESCAL, we focus on and provide solutions using the cascaded approach.

D1.4 concludes that the cascaded approach makes it possible to build IP QoS services on a global basis while only maintaining contractual relationships with adjacent operators. Hence, this approach is more scalable than the centralised approach.

Deliverable D1.4 also contains a chapter dealing further with business relationships and financial settlements between *IP Network Providers*. As service accounting, billing and marketing aspects are outside the scope of MESCAL, viability from business perspectives is addressed at the level of business relationships between NPs and related financial settlements for exchanging QoS traffic; accounting and data collection methods, charging, rating and pricing models are not addressed explicitly.

Two business cases have been identified for a MESCAL-enabled QoS-aware Internet; one for providing services based only on qualitative QoS guarantees and one for additionally providing services based on statically guaranteed quantitative QoS metrics. In both cases, services relying on hard QoS guarantees could also be provided, however not for the mass market because of scalability limitations inherent in the technical solution. The qualitative-QoS Internet business case directly corresponds to the three-tier, hierarchical model currently in place, whereas the statistical-QoS Internet business case advocates a flat Internet, where the business relationships between ISPs are of the same type, which is not affected nor dictated by the tier levels the ISPs may reside. In the flat Internet, the net flow of money always follows the flow of traffic. In the hierarchical Internet, assuming that a tier 1 ISP must always be involved, the net flow of money follows the flow of traffic until a tier 1 ISP is reached, at which point on, the net flow of money goes against the traffic.

2.2.4 Service providers

'Service Providers' offer higher-level QoS-based services encompassing both connectivity and informational aspects e.g. telephony, content streaming services. As opposed to *'IP Network Providers'*, *'Service Providers'* may not necessarily own and administer an IP network infrastructure; they need to administer the necessary infrastructure required by the provisioning of the offered services e.g. VoIP gateways, IP video-servers, content distribution servers. As such, for fulfilling the connectivity aspects of their services, *'Service Providers'* may rely on the connectivity services offered by *'IP Network Providers'*. In this sense, *'Service Providers'* interact with *'IP Network Providers'* following a customer-provider paradigm on the basis of respective agreements (SLAs). Furthermore, for expanding the geographical scope and augmenting the portfolio of the services offered, *'Service Providers'* may interact with each other on a peer-to-peer or a strict customer-provider basis.

2.2.5 Physical connectivity providers

'Physical Connectivity Providers' offer physical (up to the link layer) connectivity services between protocol-compatible equipment in determined locations. It should be noted that the connectivity services may also be offered in higher layers (layer-3 e.g. IP), however these services are mainly between specific points as opposed to the IP connectivity services offered by *'IP Network Providers'* which may be between any points in the IP address space. *'Physical Connectivity Providers'* are distinguished into two main categories according to their target market: *'Facilities Providers'* and *'Access Providers'*. These types of Providers could be seen as distinct stakeholders. One special case of a *'Facilities Provider'* is an Internet eXchange Point (IXP). An IXP is a physical network infrastructure operated by an entity with the purpose of facilitating the exchange of Internet traffic between *IP Network Provider* domains. Any *IP Network Provider* that is connected to an IXP can exchange traffic with any other *IP Network Providers* connected to the same IXP, using a single physical connection to the IXP, thus overcoming the scalability problem of individual interconnection links. Deliverable D1.4 contains a detailed discussion on the role of IXPs and their implications on the MESCAL solutions.

The services of *'Facilities Providers'* are mainly offered to *'IP Network Providers'* to provide the required link-layer connectivity in their IP network infrastructure or to interconnect with their peers as discussed previously. As such, *'IP Network Providers'* may interact with *'Facilities Providers'* following a customer-provider paradigm on the basis of respective agreements (SLAs). These interactions are analysed further from the perspective of dynamic network provisioning in Section 2.7 where a *IP Network Provider* may dynamically determine and request capacity between its IP routers from the underlying *Physical Connectivity Provider*. *'Facilities Providers'* may be differentiated according to the type of technology they rely upon (e.g. optical fibre, satellite, antennas), deployment means (terrestrial, submarine, aerial) and their size in terms of geographical span and customer base. The technological means for provisioning optical networks are reviewed in some detail in chapter 3 of deliverable D1.4 on optical network technologies and their implication on MESCAL.

The business model as described in D1.1 did not adequately capture the relationships between *'Facilities Providers'*. This is corrected in the current view. A single *Facility Provider* may interconnect the *IP Network Providers*, as in the case of an IXP, or where a single national carrier provides a leased line between them. This is depicted in Figure 2. Alternatively, private peering could be achieved through international connections through a chain of *Facility Providers*. In this case there would be separate *'Physical Connectivity Providers'* who cooperate and interwork to provide the end-to-end physical layer capability. The latter case could be captured with an additional arrow between separate *'Physical Connectivity Providers'* as shown in Figure 3.

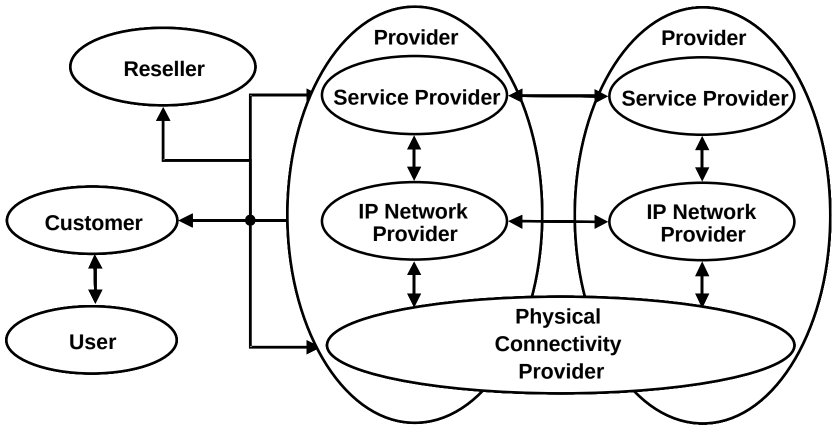


Figure 2. Revised MESCAL business model – Common Physical Connectivity Provider

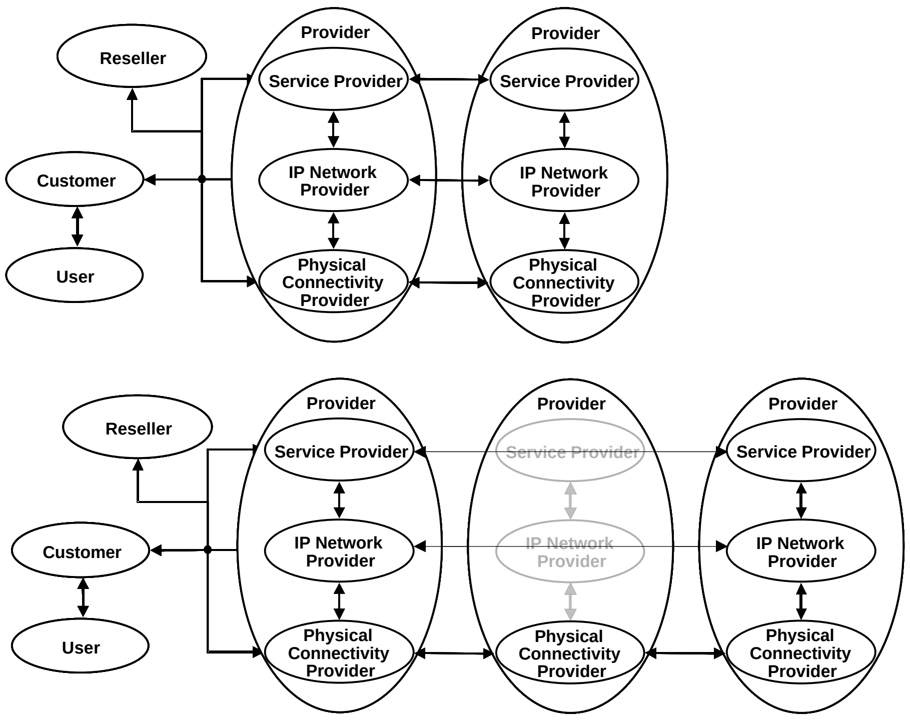


Figure 3. Revised MESCAL business model – Interworking between Physical Connectivity Providers

'Access Providers' offer services for connecting *'Customer'* premises equipment to the appropriate (*'Service'* or *'IP Network'*) *'Providers'* equipment. They own and administer appropriate infrastructure e.g. cables, concentrators. They may be differentiated according to the type of technology they employ e.g. POTS, FR, ISDN, xDSL, WLAN, Ethernet, as well as their deployment means and their size in terms of covered geographical area and customer base.

2.2.6 Resellers

'Resellers' are intermediaries in offering the QoS-based services of the *'Providers'* to the *'Customers'*. In essence, *'Resellers'* offer market-penetration services (e.g. sales force, distribution/selling points) to *'Providers'* for promoting and selling their QoS-based services in the market.

2.3 Meta-QoS classes

2.3.1 Introduction

Although there has been much work done in QoS field over the last decade, little has been undertaken to provide guidance on how to deploy QoS throughout the whole Internet. This Section is a step in this direction. It targets services that would potentially be accessible by a large Internet community independently on their network access providers' location. It doesn't deal simply with inter-domain QoS. As a matter of fact, QoS access across a limited set of ASs is not the same kind of problem as QoS access throughout the whole Internet.

We present here a new concept that is the Meta-QoS-Class concept. This concept, if it succeeds in raising general interest and agreement, is likely to dramatically help to achieve a QoS-enabled Internet.

2.3.2 Current inter-domain QoS deployment assessment

Based on current best practices, we can hardly say that QoS (if over-provisioning isn't considered as a part of QoS management) has been currently deployed inter-domain and even intra-domain in Service Providers' networks. The Internet remains an interconnection of best effort networks. The only worldwide transport service usable throughout the Internet is the best effort service. For instance there is no means for a video content provider to make it possible for their ready to pay customers to access the service via a performance guaranteed transport at large scale.

2.3.3 Requirements

An inter-domain QoS delivery solution *should* take into account some requirements that would prevent QoS techniques and architecture to impair the spirit in which the Internet has been devised since its early days. The idea is of course not to refuse any evolution in the Internet paradigm just because the Internet is as it is. The intention is to keep the features the great majority of people can agree on, because these features are deemed worth to be preserved for the good of citizens. The priority isn't necessary about technical and financial considerations. We should preserve the facility to spread Internet access, the facility to welcome new applications and the possibility to communicate from any point to any other point.

From this angle, the list of requirements *should* encompass:

- Networks *should* be ready to convey inter-domain QoS traffic before customers can initiate end-to-end SLS negotiations (just like inter-domain routing is).
- The solution *must* not, to the greatest extent possible, preclude unanticipated applications.
- A best effort route *must* be available when no QoS route is known.
- Best effort delivery *must* survive QoS.
- The solution *should* not rely on the existence of a centralised entity that have the knowledge and the control of Internet (*an Internet God*) and that doesn't exist anyway.

2.3.4 A basic QoS inter-domain problem: binding l-QC

2.3.4.1 Problem statement

A given Service and/or Network Provider offers QoS-based services to its customers. The span of these services is limited to its network boundaries. On the other hand, this Service Provider is aware that many other Service Providers, scattered in the Internet, provide also QoS-based services to their customers. From a centric view, this Service Provider wants to benefit from the QoS infrastructure made up with all the QoS-enabled networks, to expand its QoS-based services to customers outside its own administrative network.

2.3.4.2 Who is a given Service Provider going to trust?

Let's consider a QoS AS path used by clients of a given provider. This provider can have strong agreements with its neighbours, but what visibility of agreements between farther ASs? If one AS of the path does not respect its commitment, how does this provider know? Even if it knows what can it do? Even if its directly peered AS guarantees end-to-end performances and it complains to it, what will the neighbour do? Complain to the following AS? That will complain to the following AS? That will…?

The Conclusion is that the following sensible assumption has to be made: *each provider should trust only what its own peered neighbours guarantee for the crossing of their own networks*.

2.3.4.3 Using only local information to bind l-QCs

In order to provide QoS-based services, an AS implements l-QCs. Service extension to other ASs, on a low level (with regard to OSI layers), means l-QC extension outside the scope of a single AS. Then, knowing l-QCs capabilities advertised by its service peers, the basic technical question a provider has to face is: "*on what basis shall I bind my l-QC to their l-QC?*". Given one of local l-QCs what is the best match? On what criteria?

A Service Provider knows very little about agreements more than one AS hop away. These agreements can move and it's hard to have an accurate visibility of their evolutions. Therefore the provider *should* take the decision to bind one of its l-QCs to one of its AS neighbour l-QCs based solely on:

- What it knows about its own l-QCs
- What it knows about its AS neighbour l-QCs

A Service Provider *shouldn't* use any information related to what happens more than one AS hop away. It *should* try to find the best match between its l-QCs and its AS neighbour l-QCs. That is to say, it *should* bind one of its l-QC with the neighbour l-QC that has the closest performances (idea of extending l-QC). The result is that any QoS AS path is the concatenation of sheer local binding decisions.

2.3.4.4 What will ensure the AS path consistency?

At this stage, we can be confronted with a problem of QoS AS path consistency. If there's systematically a slight difference between the upstream l-QC and the downstream l-QC we can wind up with a significant slip between the first and the last l-QC. Therefore we must have a means to ensure the consistency and the coherency of a QoS AS path. The idea is to have a classification tool *that says two l-QCs can be bound together if, and only if, they are classified in the same category.* We call Meta-QoS-Class each category of this l-QC taxonomy.

From this viewpoint: *two l-QCs can be bound if, and only if, they correspond to the same Meta-QoS-Class.*

2.3.5 The Meta-QoS-Class concept

2.3.5.1 Meta-QoS-Class based on a worldwide common understanding of application QoS needs

The underlying philosophy behind Meta-QoS-Classes relies on a worldwide common understanding of application QoS needs. Wherever end-users are connected they more or less use the same kinds of applications in quite similar business contexts. They also experience the same QoS difficulties and are likely to express very similar QoS requirements to their respective providers. Globally confronted with the same customers requirements, providers are likely to define and deploy similar l-QCs, each of them being particularly designed to support applications of the same kind of QoS constraints. There are no particular objective reasons to consider that a Service Provider located in Japan would design a "Voice Over IP" l-QC with short delay, low loss and small jitter while another Service Provider located in the US would have an opposite view. Applications impose constraints on the network, independently of where the service is offered in the Internet.

Therefore, even if we strongly believe there is no Internet God, we consider that: ***There is a Customer God and he invented the Meta-QoS-Class concept.***

It should be understood that a Meta-QoS-Class is actually an abstract concept. It is not a real l-QC implemented in a real network.

2.3.5.2 Meta-QoS-Class definition

A Meta-QoS-Classes could be defined with the following attributes:

- A list of services (e.g. VOIP) the Meta-QoS-Class is particular suited for.
- Boundaries for QoS performance attributes (one-way transit delay, one-way transit variation delay –jitter-, loss rate).
- Constraints on traffic to put onto the Meta-QoS-Class (e.g. only TCP-friendly).
- Constraints on the ratio: resource for the class to traffic for the class.

Attributes could depend on AS diameter, for example a longer delay could be authorised for large Ass, and performance attributes can be weighed in order to prioritise the ones to which the service is more sensitive.

A given Meta-QoS-Class followed by the same Meta-QoS-Class should equal the same Meta-QoS-Class (invariance).

2.3.5.3 What's in and out of a Meta-QoS-Class?

Only a limited set of Meta-QoS-Classes should be defined. Each AS classifies its own l-QCs based on Meta-QoS-Class. An l-QC from an AS can be bound only with a neighbour l-QC that refers to the same Meta-QoS-Class.

A Meta-QoS-Class typically bears properties relevant to the crossing of one and only one AS. However this notion can be extended in a straightforward manner to the crossing of several AS, as long as we consider the set of AS as a super and single AS.

A Meta-QoS-Class doesn't describe the way to implement an l-QC. It is not a real l-QC. It is a classification tool for implemented l-QC.

The Meta-QoS-Class concept is very flexible with regard to new unanticipated applications. A new unanticipated application could drive a new Meta-QoS-Class. According to the end-to-end principle a new unanticipated application should have very little impact on existing l-QC, but this issue doesn't concern Meta-QoS-Classes per se, it is the problem of l-QC design and engineering.

A hierarchy of Meta-QoS-Classes can be defined for a given type of service (e.g. VoIP with different qualities). A given l-QC can be suitable for several Meta-QoS-Classes (even outside the same hierarchy). Several l-QCs in a given AS can be classified as belonging to the same Meta-QoS-Class. Private chains of peerings (outside the scope of a global reachability) can do whatever they want (not bound to the Meta-QoS-Class constraint).

The Diffserv concept of Per-Domain Behaviour (PDB) should not be confused with the Meta-QoS-Class concept. The two concepts share the common characteristic of specifying some QoS performance values. However the two concepts don't exactly overlap. The two concepts differ in their purposes. The objective for the definition of a PDB is to help implementation of QoS capabilities within a network. The objective for a Meta-QoS-Class is to help agreement negotiation between Service Providers. A PDB is closer to an l-QC than to a Meta-QoS-Class.

In summary the interest of Meta-QoS-Class concept is threefold. It:

- Gives guidance for l-QC binding.
- Allows relevant l-QC binding with no knowledge of the following distant AS agreements.
- Enforces coherency in a QoS AS path with no knowledge of the complete path.

2.3.6 The fundamental use case: the QoS Internet as a set of Meta-QoS-Class planes

We describe here what the fundamental use case, or Internet QoS model, could be, based on the Meta-QoS-Class concept. The purpose of this model is to build a QoS-enabled Internet, which keeps, as much as possible, the openness of the existing best effort Internet, and more precisely conforms to the requirements expressed above in this Section. In this model, the resulting Internet appears as a set of parallel Internets or Meta-QoS-Class planes. Each Internet is devoted to serve a Meta-QoS-Class. Each Internet consists in all the l-QCs bound in the name of the same Meta-QoS-Class. When an l-QC maps several Meta-QoS-Classes it belongs to several Internets. The user can select the Internet that is the closest to his needs as long as there is currently a path available for the destination.

We assume that in a Meta-QoS-Class plane, because we want to stay close to the Internet paradigm, all paths were to a reasonable extent, born equal. Therefore, the problem of path selection amounts to: *Do your best to find one path, as best as you can, for the selected Meta-QoS-Class plane.* This sounds like the traditional routing system used by the Internet routers. Therefore we can rely on a BGP-like protocol for the path selection process. By destination, q-BGP selects and advertises one path for each Meta-QoS-Class plane.

When, for a given Meta-QoS-Class plane, there is no path available to a destination, the only way for a datagram to travel to this destination is to use another Meta-QoS-Class plane. The only Meta-QoS-Class plane available for all destinations is the best-effort Meta-QoS-Class plane (also known as "the Internet"). There's no straightforward solution to change from one plane to another on the fly. So, there's no straightforward way to span a Meta-QoS-Class plane hole by a best-effort bridge.

This solution gives only loose administrative guarantees, however as long as all actors (especially, all service peers involved in the QoS AS path) do their job properly, the actual level of guarantee will be what is expected.

This solution stands only if l-QC "Meta-QoS-Class"-based binding is largely accepted and proceeded.

2.3.7 Proposal for a first set of meta-QoS-classes

We propose to define five Meta-QoS-Classes.

- Premium Meta-QoS-Class
- Gold Meta-QoS-Class for TCP-friendly traffic
- Gold Meta-QoS-Class for non TCP-friendly traffic
- Best effort Meta-QoS-Class
- Lower than Best Effort Meta-QoS-Class

Examples of basic groupings are:(these example are given for the sake of clarification and not to recommend a particular configuration)

- Internet with the five Meta-QoS-Classes
- Internet with only the first four Meta-QoS-Classes
- Internet with only the last two Meta-QoS-Classes

We define some parameters for each Meta-QoS-Class in the following sub-paragraphs. These parameters are: Targeted use, Performance, Constraint on the flows and Resources.

The values for the performance parameters have not been set yet. They should be derived from the knowledge of the application needs and the knowledge of the performances of the main Service Providers' networks.

2.3.7.1 Premium Meta-QoS-Class

Targeted use: mission critical applications

Performance: very low delay, very low jitter, no loss

Constraint on the flows: some sort of admission control and possibly shaping to enforce the resource requirement.

Resources: on each output interface, the traffic for the class is always much smaller than the bandwidth reserved for the class (EF based). The resources must always absorb the traffic with no loss even with bursty aggregates.

2.3.7.2 Gold Meta-QoS-Class (two classes)

Targeted use: sensitive applications split into two different classes TCP-friendly traffic and non TCP-friendly traffic. We differentiate two classes because since we allow diagram deletion a mix of TCP and non-TCP flows could put TCP flows at a disadvantage.

Performance: low delay, low jitter, low loss

Constraint on the flows: TCP friendly traffic for the TCP-friendly Class traffic.

Resources: on each output interface, the traffic for the class can be greater than the bandwidth reserved for the class (AF based) the delta has a direct impact on the loss rate.

2.3.7.3 Best Effort Meta-QoS-Class

Targeted use: current applications

Performance: no guarantees however the measured values should not be too bad

Constraint on the flows: no constraint

Resources: the ratio resource for the class to traffic for the class must not be too small.

2.3.7.4 Lower than Best Effort Meta-QoS-Class

Targeted use: any delay requirement applications

Performance: no guarantees

Constraint on the flows: services that don’t care at all about delay (may be partly because very cheap)

Resources: the resources reserved for this class must be very small compared to the other classes (included the traditional Best Effort). The ratio resource for the class to traffic for the class can be very small.

2.3.8 Future work

2.3.8.1 Thorough definition

Some work should be undertaken to refine the definition of a Meta-QoS-Class. Some parameters should be more deeply investigated. For example: how exactly should a service be described? What are the sub-attributes? How should the performance characteristics be described? With what precision? Do we need a parameter for availability? How to define it?

In the basic parameters we gave, a Meta-QoS-Class appears for a customer both as a way to convey a certain type of application (for instance video traffic) and as a way to get some guarantees in terms of one-way transit delay, one-way transit variation delay and Loss rate. It would be worthwhile to investigate in these two approaches and to decide whether we should privilege one of them or keep the two of them.

2.3.8.2 Standardising Meta-QoS-Classes?

Each Service Provider must have the same understanding of what a given Meta-QoS-Class is about. A global agreement (a.k.a. standards) is needed. This agreement could be typically reached in an international standardisation body. There must be also a mean to certify the l-QC classification made by an AS conforms to the Meta-QoS-Class standards. So the Meta-QoS-Classes standardisation effort should go along with some investigation on conformance testing requirements.

2.3.8.3 Validating a set of relevant Meta-QoS-Classes

What are the first Meta-QoS-Classes to define?

2.3.8.4 Meta-QoS-Class outclassing procedures

Additional techniques should be investigated in order the Meta-QoS-Class hierarchy to be exploited with techniques such as outclassing.

2.3.8.5 Security

Security is a main concern in a QoS-enabled Internet. Flows entering an AS and requesting QoS are likely to arrive from any AS and to be destined to any AS. So, it is of primary importance for a Service Provider to be able to filter the flows whose requests are not legitimate. Some investigations must be conducted in this direction. The Meta-QoS-Class concept opens the possibility of QoS services potentially reachable from any Internet position. Consequently, the menace of a spurious attack grows accordingly.

2.3.8.6 Finding out other use cases

If we want strong administrative guarantees we could add some mechanisms on top of the fundamental use case. One could find completely different uses of Meta-QoS-Class than the one depicted here.

2.3.9 Conclusion

In this Section we have introduced the new notion of Meta-QoS-Class that makes it possible to build a QoS-enabled Internet that will keep the main desirable properties of the existing best-effort Internet.

2.4 Review of the MESCAL Functional Architecture

This Section introduces the functionality required for the provision of inter-domain QoS services from the perspective of a single provider. The functional architecture analyses the overall problem of providing inter-domain QoS and decomposes it into a set of finer grained components. One of the objectives of this exercise is to aid the development of the MESCAL solutions by breaking it down into manageable entities while maintaining a holistic view of the overall issues to be solved. In essence it is a divide and conquer exercise.

The MESCAL functional architecture was initially proposed in deliverable D1.1 [D1.1]. This Section revisits the architecture in the light of the detailed specification activities that have subsequently taken place and culminating in the specifications in the main sections of this document. Each of the function blocks is analysed detail in Sections 3 to 7 of this deliverable, where algorithms and protocols to implement the required functionality are proposed.

2.4.1 Functional architecture overview

Figure 4 shows the initial MESCAL functional architecture developed in deliverable D1.1, showing the interactions between functional blocks at a high level. The arrows depict the direction of the main flow of information between functional blocks, generally implying a configuration or the invocation of a method in the direction of the arrow. Figure 4 also shows the interactions between providers and between customers and providers. The downstream provider on the right of the figure shows only the components directly involved in service peer interactions. An upstream provider is also implied on the left hand of the figure, although not shown explicitly. Interactions with upstream providers are a mirror of those shown with the downstream provider on the right.

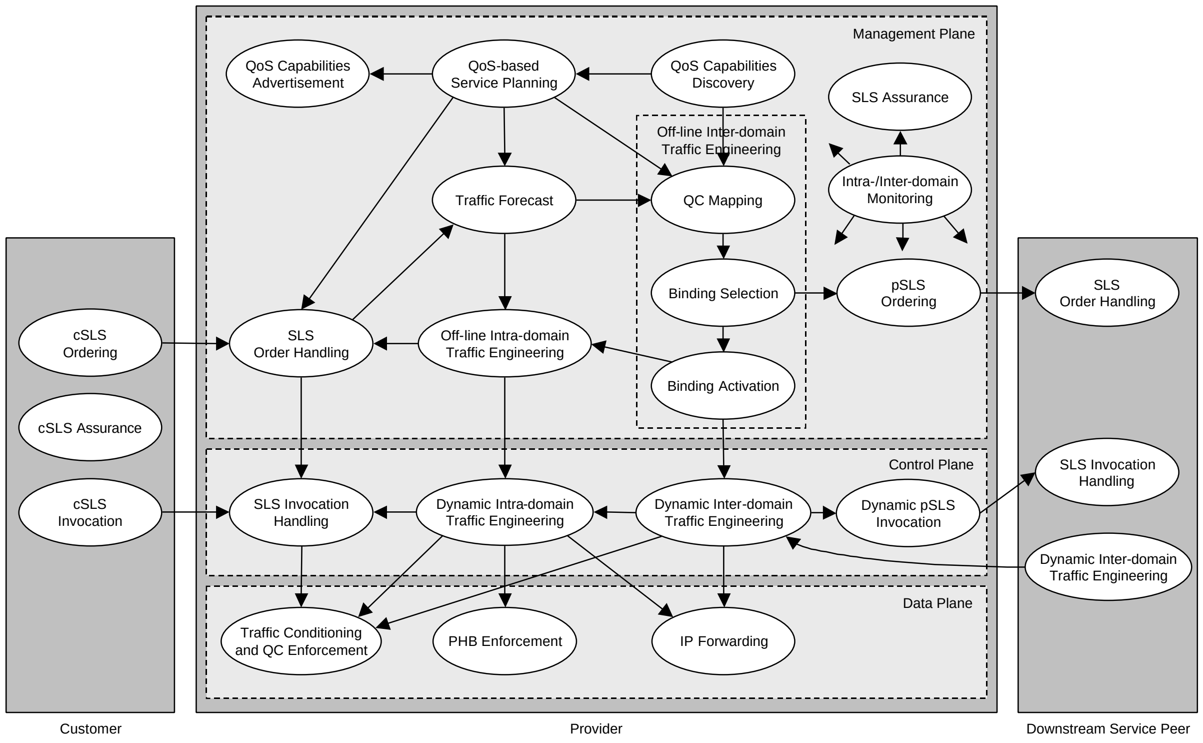

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Off-line Inter-domain Traffic Engineering
Intra-/Inter-domain Monitoring
Traffic Forecast
QC Mapping
Binding Selection
pSLS Ordering
SLS Order Handling
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Binding Activation
cSLS Assurance
Control Plane
SLS Invocation Handling
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic pSLS Invocation
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 4. The initial MESCAL functional architecture

The data plane is responsible for per packet treatment within packet arrival epochs. The control plane covers intra- and inter-domain routing, SLS invocation handling – including authentication, authorisation and admission control – dynamic resource management – including load distribution and capacity management functions. Typically, control plane functions are embedded within network equipment although they are not involved in packet-by-packet decisions.

The management plane is off-line functionality, typically located outside of the network elements in management servers. The management plane functions are responsible for planning, dimensioning and configuring the control and data planes and interacting with customers and service peers to negotiate contracts. While management plane functions are not as dynamic as control and data plane functions they are by no means static. Within the MESCAL system there is a continual background activity within the management plane at the epochs of the so-called resource provisioning cycles (RPCs). There are two RPCs in MESCAL – the intra-domain RPC which involves off-line intra-domain Traffic Engineering, and the inter-domain RPC which involves off-line inter-domain Traffic Engineering. The latter may be further decomposed into a *Binding Selection Cycle* and a *Binding Activation Cycle* (see Section 5.3.2). The RPCs aim at proactively optimising network resources to meet predicted demand and to build in sufficient spare capacity to avoid the burden of reconfiguring the network for each and every SLS subscription or renegotiation, without the inefficiencies and costs associated with massively over provisioning resources.

While the architecture describes the full set of functions required for a provider to participate in the end-to-end provision of QoS-based IP services by no means does it prescribe the implementation means by which they will be realised – within network equipment or in external management servers, with automated or manual processes. This is a matter for each provider. While the full set of functional blocks (or their equivalent) are expected to be in place in downstream providers, MESCAL does *not* assume that *automated* processes will always implement all blocks. This deliverable proposes algorithms suitable for deployment as automated processes in the traffic engineering and service management functional blocks but it is also possible to deploy much of the management plane through manual processes, at the cost of reduced responsiveness or flexibility. For some of the service options identified in this deliverable, the algorithms or manual processes required to implement the functionality might be trivial. For instance, the *loose guarantees service option* does not require explicit admission control functionality in the SLS Invocation Handling block, and the QC Mapping, Binding and Activation processes are simplified due to its adoption of well-known Meta-QoS-Classes and the restriction to bindings only with the same Meta-QoS-Class in service peer domains.

The following subsections identify the major aspects of the functionality contained within each of the blocks shown in Figure 4 and highlight the changes to the functional architecture that have been made since D1.1.

2.4.2 QoS-based Service Planning, QoS Capabilities Discovery and Advertisement

QoS-based Service Planning encompasses all the higher level business related activities responsible for defining the services that the provider should offer to its customers and service peer providers. These are specified according to the business objectives of the provider, and include l-QCs within the scope of its own network and e-QCs combining its local QoS-based services with those offered by its service peers.

Prior to any pSLS agreement with a neighbouring provider, a provider discovers the QoS capabilities, capacities, destination prefixes and costs of potential service peer providers thanks to the *QoS Capabilities Discovery* functional block. Once l-QCs and e-QCs have been defined and engineered (by Intra- and/or Inter-domain TE) the *QoS Capabilities Advertisement* block is responsible for promoting the offered services so that its customers and service peer providers are aware of its offerings. It is envisaged that a variety of advertising means will be used, ranging from digital marketplaces or other automated peer-to-peer processes to conventional techniques such as salespersons, newspapers and word of mouth.

2.4.3 Off-line Traffic Engineering

Traffic Forecast is responsible for aggregating and forecasting traffic demand. During a provisioning cycle, the set of subscribed cSLSs and pSLSs are retrieved from *SLS Order Handling* and an aggregation process derives a traffic matrix with the demand per ordered aggregate between ingress and egress points of the domain (ASBRs). The demand matrix is used by the intra- and inter-domain traffic engineering processes to calculate and provision the local and inter-domain resources needed to accommodate the traffic from established SLSs as well as those anticipated to be ordered during the provisioning cycle.

Binding Selection is the process of combining l-QCs of the local domain with o-QCs of other domains, learned through *QoS Capabilities Discovery*, to construct potential e-QCs that meet the service requirements defined by *QoS-based Service Planning*. It should be noted that *Binding Selection* might result in a number of QoS-bindings for a given e-QC. QoS-bindings with the same service-peering provider may differ in the l-QC and subsequently in the o-QC they use. Alternatively, QoS-bindings may differ when established with different service-peering providers.

Binding Activation is responsible for mapping the predicted traffic matrix to the inter-domain network resources (once pSLSs have been established), satisfying QoS requirements while aiming at optimising the use of network resources across AS boundaries. *Binding Activation* decides which of the established QoS-bindings will be *put in effect* in the network for implementing an e-QC together with the associated routing constraints for those e-QCs. The QC-bindings in effect will be enforced through routing decisions as well as configurations of the *Traffic Conditioning and QC Enforcement* block, e.g. configuring the egress ASBR to perform DSCP remarking for realising a QC-binding. The latter configuration can be made directly to the egress router or passed through *Dynamic Inter-domain Traffic Engineering*.

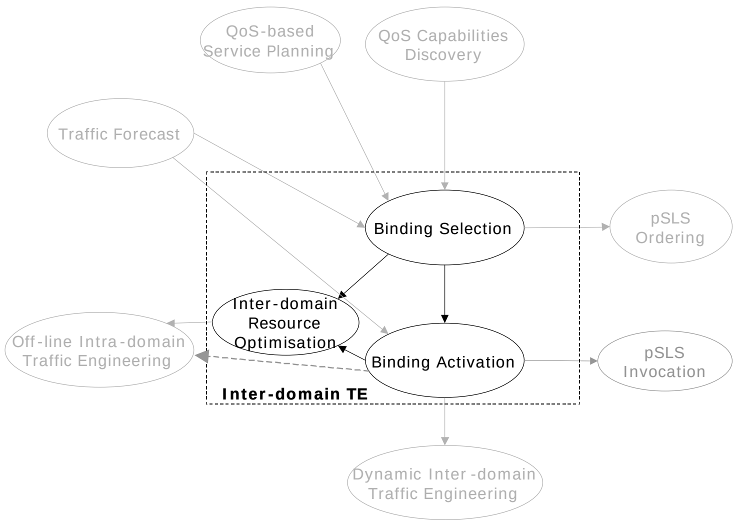

QoS-based Service Planning
QoS Capabilities Discovery
Traffic Forecast
Binding Selection
pSLS Ordering
Inter-domain Resource Optimisation
Off-line Intra-domain Traffic Engineering
Binding Activation
pSLS Invocation
Inter-domain TE
Dynamic Inter-domain Traffic Engineering

Figure 5. Decomposition of the Offline inter-domain TE

Although in the initial Functional Architecture described in D1.1 [D1.1] the inter-domain Traffic Engineering system was decomposed into the *QC Mapping*, *Binding Selection* and *Binding Activation* blocks, after a more detailed study of the functionality of the blocks as well as the corresponding algorithms, a change in this decomposition was decided which is depicted in Figure 43. First, the functionality of the *QC Mapping* functional block was considered too lightweight to justify a single functional block and was incorporated in the *Binding Selection* block as the first step of its algorithm. Moreover, both *Binding Selection* and *Binding Activation* have to run an optimisation algorithm, which will decide on the most optimal resource allocation in terms of inter- as well as intra-domain cost in order to satisfy a predicted traffic demand. This resource allocation could be either for the establishment of the pSLSs for the next *Binding Selection* period or for the allocation of the inter-domain resources for the next provisioning cycle.

Consequently, we have included in the Traffic Engineering System an *Inter-domain Resource Optimisation* block, which realises the algorithm described above and is called both *by Binding Selection* and *Binding Activation*. Of course, the input to the algorithm will be different when called by *Binding Selection* and when called by *Binding Activation* since in the first case a traffic demand for a longer period will be passed as input to the algorithm while in the latter case a shorter term prediction of the traffic demand will be the input.

Moreover, when Binding Activation triggers the *Inter-domain Resource Optimisation* algorithm the allocation of resources is constrained by the already established pSLSs while *Binding Selection* has to consider different hypothetical scenarios of pSLSs in order to decide which one of them leads to a more optimal solution in terms of resource utilisation and at the same time satisfying the traffic demand.

Off-line Intra-domain Traffic Engineering computes the intra-domain network configuration in terms of routing constraints and PHB capacity requirements in order to satisfy the predicted traffic demand at intra-domain RPC epochs.

The off-line intra-domain TE block has been further decomposed into two sub-components: *Resource Optimisation* and *Network Reconfiguration Scheduler.* The *Network Reconfiguration Scheduler* is the control system for the Offline Intra TE block. It has two main purposes, handling computation requests to *Resource Optimisation* (Resource Provisioning Cycles, Inter-domain Traffic Engineering "what if" queries, etc) and scheduling the reconfiguration of the network using link weight settings computed by the *Resource Optimisation* block. The *Resource Optimisation* block contains the OSPF link weight optimisation algorithm. It is a passive block, until called by the *Network Reconfiguration Scheduler* at which point it collects a traffic demand matrix and a network topology and computes an optimal set of link weights. Computed weights are deposited in a link weight database inside the Offline Intra-domain Traffic Engineering block, until they are put into operation in the network by the *Network Reconfiguration Scheduler.*

The interactions required between off-line inter- and intra-domain TE and the options for coupling/decoupling the inter- and intra-domain RPCs are analysed in detail in Section 5.

2.4.4 Dynamic Traffic Engineering

Dynamic Inter-domain Traffic Engineering runs within an inter-domain RPC and is responsible for inter-domain routing e.g. qBGP advertisement, qBGP path selection and for dynamically performing load balancing between the multiple paths defined by the static component based on real-time monitoring information changing appropriately the ratio of the traffic mapped on to the inter-domain paths.

Dynamic Intra-domain Traffic Engineering is the dynamic management layer as defined in TEQUILA [TEQUI]. This includes the intra-domain routing algorithms, e.g. QoS-enhanced OSPF, together with other dynamic algorithms to manage the resources allocated by *Off-line Intra-domain Traffic Engineering* during the system operation in real-time, in order to react to statistical traffic fluctuations and special arising conditions within an intra-domain RPC. It basically monitors the network resources and is responsible for managing the routing processes dynamically as well ensuring that the capacity is appropriately distributed among the PHBs.

2.4.5 SLS Management

The SLS Management functionality can be split into two parts: (a) the part responsible for the contracts offered by the provider to its customers, i.e. the end-customers and interconnected providers, and (b) the part responsible for the contracts requested by the provider from its peer providers. The resulting functional components are named "*SLS Order Handling*" and "*SLS Ordering*" respectively. While the ordering process establishes the contracts between the peering providers, the invocation process is required to commit resources before traffic can be exchanged, with "*SLS Invocation Handling*" and "*pSLS Invocation*" providing the necessary functionality.

SLS Order Handling is the functional block implementing the server side of the SLS negotiation process. Its job is to perform subscription level admission control. The *Off-line Intra-domain Traffic Engineering* block will provide *SLS Order Handling* with the resource availability matrix (RAM) which indicates the available capacity of the engineered network to accept new SLS orders – both within the AS and on any inter-domain pSLSs it has with neighbouring ASs. *SLS Order Handling* will negotiate the subscription of both cSLSs and pSLSs – they will be (largely) treated in the same way. *SLS Order Handling* maps incoming SLS requests onto the o-QCs it can offer and investigate whether there is sufficient intra- and inter-domain capacity, based on the RAM for that o-QC.

pSLS Ordering is the client side of the pSLS negotiation process. During an inter-domain RPC *Binding Selection* may identify the need for new pSLSs with service peers. *pSLS Ordering* implements the decisions of the *Binding Selection* algorithms and undertakes the negotiation process.

The *pSLS Invocation* function block is responsible for invoking pSLSs with peer domains. The pSLSs have already been subscribed through an ordering process between *pSLS Ordering* and *SLS Order Handling*. Optionally *pSLS Invocation* may be directly invoked by *Dynamic Inter-domain Traffic Engineering* to cater for fluctuations in traffic demand which are significantly different to those forecasted and used by *Binding Activation* for the current RPC. Whether or not this should trigger a new binding activation cycle by involving *Binding Activation* and *Inter-domain Resource Optimisation* is a topic for further study.

Admission control is needed to ensure that the network is not overwhelmed with traffic when the network adopts a policy of overbooking network resources at the subscription level. *SLS Invocation Handling*, containing the admission control algorithm, receives signalling requests from customers or peer providers for cSLS and pSLS invocations respectively. *SLS Invocation Handling* checks whether the invocation is conformant to the subscribed SLS and whether there is sufficient capacity in the local AS and also on the inter-domain pSLSs in the case of SLSs that are not terminated locally.

2.4.5.1 Monitoring and SLA Assurance

Monitoring is responsible for both node and network level monitoring through both passive and active techniques. It is able to collect data at the request of the other functional blocks and asynchronously notify the other functional blocks when thresholds are crossed on both elementary data and derived statistics.

For simplicity in the diagram the full set of interactions with *Monitoring* is not depicted, however *SLS Invocation Handling*, *Dynamic Inter-/Intra-domain Traffic Engineering* and *pSLS Invocation* blocks continually use monitored data in order to operate. The less dynamic *Off-line Inter-/Intra-domain Traffic Engineering* functions as well as *Traffic Forecast* use monitored network statistics at RPC epochs. *Traffic Forecast* uses historical data to improve the accuracy of future traffic matrix estimates.

Inter-domain monitoring could take several forms: monitoring inter-domain links (pSLS) only; monitoring end-to-end performance across several ASs through loop-backs or remote probes for one-way measurements; collection of data generated by service peers (possibly through BGP advertisements, or through another monitoring data exchange protocol). Alternatively third part auditing may be a more acceptable means for both monitored and monitoring ASs.

SLS Assurance compares monitored performance statistics to the contracted QoS levels agreed in the SLSs to confirm that the network or service peer-networks are delivering the agreed service levels.

2.4.5.2 Traffic Conditioning and QC Enforcement, PHB Enforcement and IP Forwarding

Traffic Conditioning and QC Enforcement is responsible for packet classification, policing, traffic shaping and DSCP marking according to the conditions laid out in previously agreed SLSs and the invocation of those SLSs. At ingress routers the *Traffic Conditioning* function is responsible for classifying incoming packets to their o-QC and subsequently mark them with the correct DSCP for the required l-QC. At the egress router the *QC Enforcement* function may need to remark outgoing packets with the correct DSCP as agreed in the pSLS with the service peer. In other words *QC Enforcement* is responsible for implementing the data-plane binding from l-QC to o-QC of the service peer. Note that *QC Enforcement* is not responsible for selecting the correct peer AS: this is decided by qBGP (part of the *Dynamic Traffic Engineering* blocks in Figure 4), therefore *QC Enforcement* does not implement the full QC mapping/binding process in the data plane.

PHB Enforcement represents the queuing and scheduling mechanisms required to be present in order to realise the different PHBs with the appropriate configuration as defined by the TE related blocks.

IP Forwarding represents the functionality needed to forward IP datagrams based on the information maintained in the corresponding FIBs. Optionally, IP forwarding may also include mechanisms to perform multipath load balancing.

2.4.6 Interactions between SLS Management and Dynamic Inter-domain Traffic Engineering

This Section describes the relationship between *SLS Management* and qBGP *when we have an agreement either for a new or an updated pSLS*. Note that these interactions are not the *only* interactions between the MESCAL management functions and qBGP, for example traffic engineering decisions will also control and influence the qBGP machinery.

The rest of this Section is organised as follows. We review briefly the structure of pSLSs and the functionality of qBGP. This review is at an abstract level, since pSLSs and qBGP are defined in Sections 4 and 5.5.1 respectively. The second part of this Section is devoted to the actual exchange of information between pSLS and qBGP. We discuss **where, what**, **when** and **who** is responsible for the information exchange.

2.4.6.1.1 Review of pSLS

A pSLS contains the following constituents that have been agreed between two ASs as part of the *SLS Order Handling* function:

- A defined offered QoS Class, o-QC (required for all solution options);
- Reachability information: a set of destination addresses to which this o-QC is valid (required for statistical and hard solution options; not required for loose solution option);
- A bandwidth (i.e. a data rate, in units of bits/second) that defines the rate at which, traffic may be sent within the terms of this pSLS, possibly including a traffic profile (required for statistical and hard solution options; not required for loose solution option);
- Time schedule (required for all solution options).

It is anticipated that in a case where there are multiple links between two ASs, then for each link we will in general have different values for some of the constituent parameters enlisted above. For example, the bandwidth may be different, or the reachable address prefixes may be different for different peering links. This is addressed by assigning separate pSLSs to each link.

2.4.6.1.2 Review of qBGP

qBGP will perform inter-domain path selection based on QC-related information and path availability information. As described in Section 5.5.1, qBGP allows exchange of QoS Service Capabilities, QC identifier, and QoS performance characteristics.

2.4.6.2 Interactions

2.4.6.2.1 Introduction: principal entities in pSLS-qBGP interaction

The pSLS – qBGP interaction is illustrated with the pair of autonomous systems shown in Figure 6. Each AS contains a management node, denoted X and Y respectively (we assume one per AS; discussions of backup nodes are outside the scope of this discussion). For the pSLS agreement between AS1 and AS2, X is responsible for performing the *pSLS Ordering* function, and Y is responsible for the *SLS Order Handling* function. Nodes X and Y are thus responsible for agreeing the pSLS (or pSLSs) between AS1 and AS2.

The other entities in scope here are:

- Upstream AS ingress node(s) (i.e. A in Figure 6);
- Upstream AS egress node(s) (i.e. B in Figure 6);
- Downstream AS ingress node(s) (i.e. C in Figure 6).

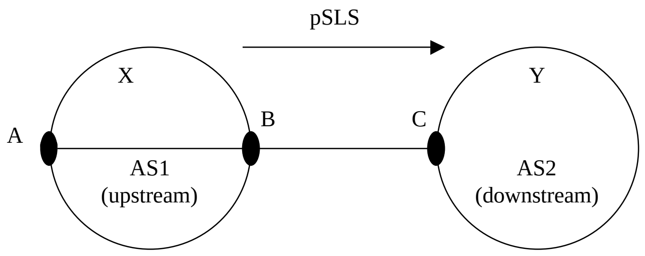


Figure 6. Two adjacent autonomous systems

Between B and C there is an exterior qBGP protocol flow (e-qBGP) and between A and B there is an interior qBGP (i-qBGP) session. We assume that a pSLS has just been agreed (either a new or a revised old one) between AS1 and AS2. In the following we will elaborate on the interaction required between the management functions and qBGP, based on the following: *where* do we foresee this interaction, *what* is the information included in that interaction, *when* this interaction happens, and finally *who* is responsible to perform that interaction.

2.4.6.2.2 Where

Routing advertisements are propagated from AS2 to AS1 using e-qBGP. These advertisements must include some QoS information that is part of the agreed pSLSs between the two ASs. In general, the two domains will filter out any other advertisement that is not part of an agreement. Thus, after a pSLS is agreed, whether new or revised, both parties should enable the exchange of such advertisements.

We therefore conclude that interaction between pSLS information and qBGP is required at the following locations:

- At the ingress nodes of the downstream AS (i.e. C in Figure 6), to implement a policy that enables the related qBGP advertisements towards the upstream AS;
- At the egress nodes of the upstream AS (i.e. B in Figure 6), to implement a policy that allows (stops filtering out) the related qBGP advertisements.

Additionally, when a new pSLS is agreed, the upstream node within the AS (i.e. A in Figure 6) has to know about the new available resources in order to use them in the egress selection process. The Interior qBGP (i-qBGP) within an AS, between A and B in our example, will provide the appropriate reachability and QoS information. If the domain's approach is that bandwidth information is not carried in i-qBGP then there are two ways for the internal nodes, like A, to "learn" that information. Either we run an IGP with Traffic Engineering (TE) extended LSAs including inter-domain links as TE-links (as proposed in [Vass03]), or the management node X could pass the pSLS bandwidth information directly to ingress node(s) A.

2.4.6.2.3 What

Having identified *where* the information is exchanged, we will now look into *what* is the required information to be exchanged.

2.4.6.2.3.1 Policy filters

As outlined in Section 2.4.6.2.2 the downstream AS must advertise qBGP reachability information to the specific addresses included in the pSLS or to "all addresses" in the case where reachability information is not specified in the pSLS. Thus the appropriate policies that allow these advertisements should be conveyed to the downstream AS ingress nodes (C in our example). Similarly, the upstream AS must allow these advertisements to be accepted and not filtered out, and further allow them to propagate into i-qBGP after applying the path selection algorithm. Therefore the appropriate policy for allowing in (i.e. stop filtering out) the related advertisements should also be downloaded to the qBGP process in the appropriate node(s), router B in our example.

2.4.6.2.3.2 QoS attributes

Routing advertisements are propagated from AS2 to AS1 using e-qBGP. These advertisements must include some QoS attribute. This QoS information is closely related to the pSLSs agreed between the two ASs. Thus, the information passed from a management node to qBGP must be the agreed o-QC. Note that this does not necessarily mean that the o-QC is the actual attribute included in qBGP, but rather that the qBGP advertised information needs to be related somehow with the agreed o-QC. The only requirement for this relationship is that the o-QC values must be the worst-case upper bound for the relevant qBGP QoS attribute values, thus allowing some flexibility in what is actually advertised into qBGP. The decision of the actual parameters that constitute the qBGP QoS attribute are for further study: for example, in a simple case we can just copy the appropriate o-QC values into the QoS attribute fields, and still be compliant with worst-case upper bound requirement.

This o-QC information is also required for the policy filters described in Section 2.4.6.2.3.1, and therefore o-QC is required at both the upstream AS egress nodes and the downstream AS ingress nodes.

2.4.6.2.3.3 Reachability information

Reachability information, i.e. specific address prefixes, is required both as part of the policy filter information and also for injection into qBGP. For the former reason, it is therefore required at both upstream AS egress nodes and at downstream AS ingress nodes. If the information about specific address prefixes is not part of the pSLS agreement, then it is assumed to be “wildcard”, that is the equals all the address prefixes to which there is reachability with the best-effort class.

2.4.6.2.3.4 Bandwidth

As discussed in the last paragraph of Section 2.4.6.2.2, bandwidth availability on the egress link for a particular QC is required for TE functions within the upstream AS, i.e. AS1 in our example. One of the TE functions that require this information is the egress path selection process of the ingress nodes of the upstream AS (e.g. node A). In Section 2.4.6.2.2 we described a number of alternatives of how this information becomes available to ingress nodes, and one of these alternatives included using qBGP as that means. In the rest of this Section we will assume that the preferred alternative is qBGP, and will discuss how and where this bandwidth must be injected into qBGP.

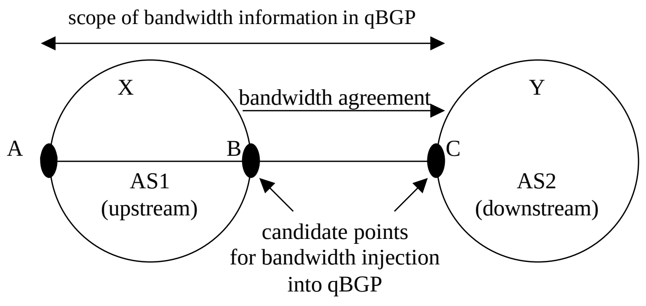


Figure 7. The case of bandwidth in qBGP

If qBGP is used to propagate pSLS bandwidth within the upstream domain, the scope of this propagation is **only** between the ingress node of the downstream AS, i.e. node C, and all the ingress nodes of the upstream AS, e.g. node A, see Figure 7. There are two principal alternatives as to where bandwidth is injected into qBGP if this policy is adopted. One is at the egress point of the upstream AS of the agreement, i.e. node B in the example, and the other alternative is at the ingress node of the downstream AS, i.e. node C of our example.

We propose to choose the latter alternative for two reasons. First, because node C already is responsible for setting the QoS attributes of the qBGP advertisements towards node B. Second, this alternative gives us the ability to perform dynamic TE with e-qBGP at node B, in addition to the TE for egress selection with i-qBGP at node A (see Figure 8). Figure 8 extends our model to the case of multiple links between the upstream and downstream AS: node B can use bandwidth information propagated using e-qBGP to select either path BC or BE.

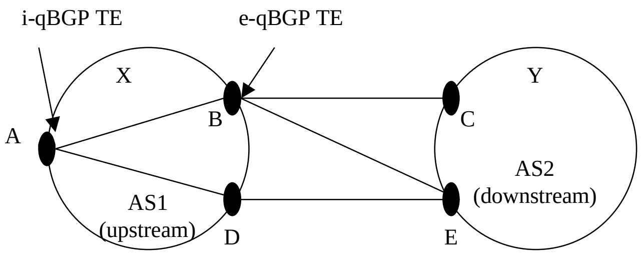


Figure 8. Illustration of i-qBGP and e-qBGP dynamic traffic engineering

2.4.6.2.3.5 Summary

Table 1 summarises **what** information needs to be passed from the pSLS related functions to qBGP:

Upstream AS egress nodes (e.g. B)	**Downstream AS ingress nodes (e.g. C)**
Policy Filter (**allow in**) for qBGP advertisements	Policy Filter (**allow out**) for qBGP advertisements
o-QC	o-QC
Reachability (destination addresses)	Reachability (destination addresses)
	Bandwidth

Table 1. Summary of data transferred from pSLSs to qBGP

2.4.6.2.4 When

The information related to a pSLS that needs to be exchanged between the management functions and qBGP, as identified in Section 2.4.6.2.3, needs to be conveyed to the qBGP machinery each time:

- A pSLS is created, modified or deleted (i.e. part of the *SLS Order Handling* function); or
- Some bandwidth is dynamically invoked within the given pSLS as part of the *SLS Invocation Handling* function (based on the restrictions discussed in the last paragraphs of Sect. 2.4.6.2.2 and 2.4.6.2.3).

2.4.6.2.5 Who

The interactions discussed in this Note are between the pSLS related blocks and the (edge) nodes that support qBGP. In the MESCAL functional architecture [D1.1] these are *pSLS Ordering* and *pSLS Invocation* in the case of an upstream AS (e.g. AS1 in Figure 1) and *SLS Order Handling* and *SLS Invocation Handling* in case of a downstream AS (e.g. AS2 in Figure 1).

The offline SLS management blocks are assumed to reside in some management server nodes, X and Y in our example, while the dynamic functions regarding the pSLS invocations are implemented in the edge routers, B and C in the example. The communication between the SLS management nodes and the qBGP routers can be implemented using any standardised management protocol e.g. SNMP or any other proprietary means e.g. Telnet/CLI.

2.4.7 Other functions and capabilities

The functional architecture covers those capabilities necessary for deploying and operating inter-domain QoS services. A provider may need other more general support functions such as fault and configuration management, but as these are not an explicit part of the inter-domain QoS provision problem they are not covered in this architecture. The role of dynamic network provisioning and the role of a network provisioning cycle is analysed in Section 2.7. As a result a *Network Planning* block has been added to Figure 9 in Section 2.4.8 to demonstrate where this functionality is positioned. It should be noted that this study was limited to the level of updating the functional architecture. Detailed studies of interactions between *Network Planning* and underlying *Physical Connectivity Providers* or algorithms for optimising the deployment of *physical* resources, e.g. optical networks, are out of scope of MESCAL.

It is envisioned that rather than being entirely hard-coded at development or installation time, the behaviour of many of the MESCAL functions and algorithms can be influenced at run time by a *Policy Management* infrastructure. Policies are expected to cover the *SLS Management* and *Traffic Engineering* functional blocks. There are no explicit functional blocks shown to handle multicast services. As described in Section 2.8 it is assumed that multicast functionality distributed over several of the blocks and only two additional blocks have been identified: *Dynamic Group Management* and *RPF Checking*. These are introduced in Section 2.8 and are discussed in detail in Section 7.

For most providers, an important aspect of providing service differentiation is the means for charging appropriate rates for different service levels. Metering, rating, billing and other commercial aspects of QoS delivery are outside of the scope of MESCAL and are therefore not part of the specified functionality. The issues associated with financial settlements according to the various business models for interactions between network providers have been studied, however, and an analysis of the implications on the MESCAL solutions is documented in deliverable D1.4 [D1.4].

2.4.8 Updated Functional Architecture

Following the discussions in the previous subsections, Figure 9 shows the current view of the MESCAL functional architecture, which forms the foundation of the MESCAL solutions and the detailed per-component sections in the remainder of this document.

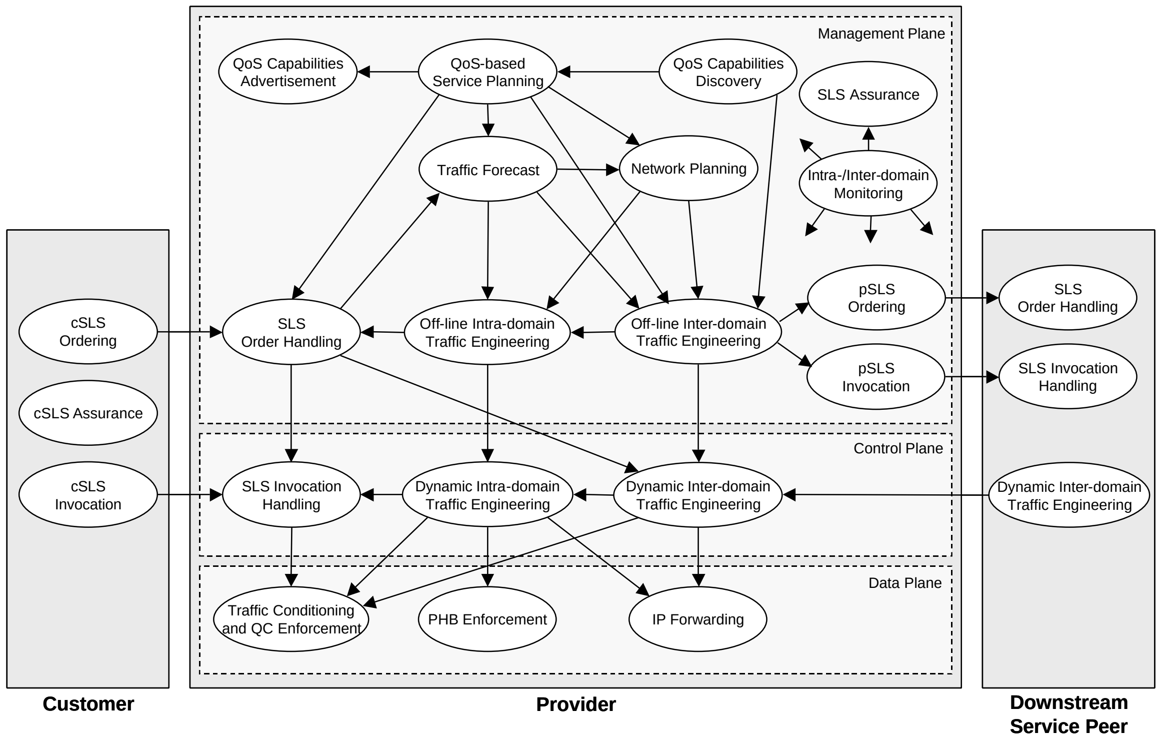

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Traffic Forecast
Network Planning
Intra-/Inter-domain Monitoring
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering
pSLS Ordering
pSLS Invocation
SLS Order Handling
SLS Invocation Handling
cSLS Assurance
Control Plane
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 9. Updated functional architecture

As a summary, the main changes are as follows:

- *pSLS Invocation* has been clearly positioned in the management plane as the component implementing the decisions of *Binding Activation*.
- Additional interactions have been specified between *SLS Order Handling* and the qBGP processes contained within *Dynamic Inter-domain TE.*
- A *Network Planning* block has been included to position the role of dynamic network provisioning within the MESCAL functionality.
- *Off-line Inter-domain TE* and *Off-line Intra-domain TE* have both been further decomposed as highlighted in the previous discussions and justified in detail in the main sections of this deliverable. For simplicity, both groups of functions are shown as a single functional block in the overall functional architecture.

2.5 Interoperability of MESCAL service options

2.5.1 Introduction

Several MESCAL service options have been defined in the deliverable D1.1. Each service option [D1.1] provides different QoS-based services guarantees and is supported by a dedicated solution option. . Part of the technical means and protocols used to implement and to deploy each solution option can sometimes differ. In the contrary, some of these solution options can make use of common techniques and/or protocols, but their use can vary depending on the context of each individual solution option (for example a dynamic inter-domain protocol could be used for all solution options but the information to be carried by this routing protocol messages could differ). These differences can be sensitive and can become critical when deploying more than one solution option within the same AS (*in the rest of this Section, we will refer to this as the Co-existence Scenario*) or when extending the scope of a given solution option through an AS supporting a different solution option (*in the rest of this Section, we will refer to this as Inter-working Scenario*). Interoperability issues raised during this study together with potential solutions for solving them will very likely introduce new requirements that in turn will impact the solution option themselves including protocols and algorithms. Both service and technical considerations are taken into account when dealing with these co-existence and the inter-working scenarios.

This Section summarises the issues discussed within the chapter 4 of the D1.4 deliverable [D1.4] which develops those co-existence and inter-working scenarios in more details. Major problems raised during the study of these aforementioned scenarios have been highlighted. Only major issues are presented and restated here. No arguing is developed in this Section. A summary of the recommendations proposed in D1.4 is also provided.

2.5.2 Service considerations

The main purpose of these service considerations section is to qualify and classify the equivalent service option resulting from the interconnection of two ASs operating different service options, independently of any technical inter-working considerations. Only inter-working service scenarios providing an upstream service options with a better service (in terms of guarantees and not the QoS performance characteristics) are considered. This service evaluation phase will decrease the number of possible scenarios that need to be studied deeply from a technical angle.

This classification effort leads to the conclusion that only the following scenarios are valid from an inter-working service perspective:

- Extending the loose service option through the statistical service option,
- Extending the loose service option through the hard service option,
- Extending the statistical service option through the hard service option.

This service logic isn't strictly respected within the technical discussions. The motivation is to be able to address transit scenarios (*the transit scenario could be defined as follows: a given AS that enables a service option x could cross one or more ASs that offers different service option y in order to join remote service option x clouds*) and traffic bi-directionality issue.

2.5.3 Co-existence scenario

This scenario consists at examining the implications of the existence of several service options in the same autonomous system. For this purpose we examine the impact of the deployment of each service option on the network infrastructure and we qualify the compatibility of these functions between service options (For more details refer to [D1.4], Chapter 5). The basis of this comparison relies upon the technical description of the three solution options provided in D1.1.

The four discussed scenarios are:

- Co-existence of the loose and the statistical solution options in the same AS,
- Co-existence of the loose and the hard solution options in the same AS,
- Co-existence of the statistical and the hard solution options in the same AS,
- Co-existence of the all solution options within the same AS.

The discussion about the above scenarios focuses on the following:

- Main technical divergence issues between the considered solution options,
- A brief description of the problems raised,
- A list of recommendations to fix these problems,
- The adopted solution(s).

	Subjacent concepts in conflict	*Encountered problems*	*Recommendations*	*Adopted solution(s)*
Co-existence of the loose and the statistical solution options in the same AS	• Use of meta-QoS-class concept • Use of q-BGP • Bandwidth management • Contractual guarantees • Information contained in pSLS	• Differentiate the intra-domain path and the egress point per solution option • Usage of routes learned via q-BGP • Usage of common and shared network infrastructure for both solution options (Multiple Solution Option Management Problem, MSOMP)	• Use different ranges of DSCP values for the two solution options. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure. • The q-BGP process must have a means to separate announcements per solution option so that it can process each announcement according to the service option it belongs to.	• Use different ranges of DSCP values for the two solution options. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure. • The q-BGP process must have a means to separate announcements per solution option so that it can process each announcement according to the service option it belongs to.
Co-existence of the loose and the hard solution options	• Use of q-BGP	• Usage of routes learned via q-BGP for the two solution options. Possible routing inconsistencies, inefficiency of inter-PCS communications.	• Differentiate q-BGP updates per service option. • At a peering point, the activation of the hard service option must be conditioned by the activation of the loose service option. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure.	• Differentiate q-BGP updates per service option. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure.
Co-existence of the statistical and the hard solution options,	• Use of meta-QoS-class concept • Use of q-BGP • Information contained in pSLS	• Usage of common and shared network infrastructure for both solution options • Usage of routes learned via q-BGP	• Use different ranges of DSCP values for the two solution options. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure. • Differentiate q-BGP updates per service option.	• Use different ranges of DSCP values for the two solution options. • Build a management system able to handle simultaneously the two solution options on top of a common and shared network infrastructure. • Differentiate q-BGP updates per service option.
Co-existence of all solution options	• Use of meta-QoS-class concept • Use of q-BGP • Information contained in pSLS	• Differentiate the intra-domain path and the egress point per solution option • Usage of routes learned via q-BGP • Usage of common and shared network infrastructure for all solution options	• Use a dedicated range of DSCP values for each solution option. • Build a management system able to handle simultaneously all solution options on top of a common and shared network infrastructure. • Differentiate q-BGP updates per service option.	• Use a dedicated range of DSCP values for each solution option. • Build a management system able to handle simultaneously all solution options on top of a common and shared network infrastructure. • Differentiate q-BGP updates per service option.

2.5.4 Inter-working scenario

The inter-working scenario deals with technical problems encountered when extending a given solution option through an AS offering different solution option(s). Thus, the following scenarios have been studied:

- Extending the loose service option through the statistical service option,
- Extending the loose service option through the hard service option,
- Extending the statistical service option through the hard service option.

The following table highlights the major problems:

	Encountered problems	*Bi-directionality problems*	*Recommendations*	*Adopted solution(s)*
Extending the loose service option through the statistical service option	• What is the methodology for inserting an o-QC in a given meta-QoS-class? • Who will adapts the q-BGP announcements • Usage of routes learned via q-BGP: if no indication is inserted in q-BGP messages, any solution option could pretend that this route is valid from a service point of view. As a result, this could generate black holes in the Internet. • The bandwidth management isn't optimal since: in the loose solution option side policing is done per meta-QoS-class and in the statistical solution option shaping is achieved on a per pSLS basis.	• How to transform a route learned from a loose service option in an o-QC that remains compatible with the solution option? • Differentiate the o-QC that are built thanks to a pure statistical solution option pSLS, and the ones that are bought from an AS offering loose solution options	• Specify a methodology the statistical solution option should follow in order to adapt o-QCs to the meta-QoS-class concept of the loose solution option • Solve the bandwidth management problem • When the two solution options need to coexist in the same AS: • Differentiate q-BGP announcements per solution option • Use different range of DSCP per option for a given PDB. • Deploy the loose solution option when the statistical one is offered	• Differentiate q-BGP announcements per solution option
Extending the loose service option through the hard service option	• Usage of routes learned via q-BGP: if no indication is inserted in q-BGP messages, any solution option could pretend that this route is valid from a service point of view. As a result, this could generate black holes in the Internet.		• Make mandatory the deployment of the loose solution option when the hard solution option is offered • Adopt a single channel signalling: this consists in introducing a dedicated flag in q-BGP messages that will indicate the presence of "hard solution option 3" holes in a given path. • Adopt the double signalling channel. This could be achieved in at least two ways: • Duplicate the q-BGP announcements and indicate the service option it serves • As far as the hard solution option is considered, an AS will announce only its PCS thanks to the use of an identifier (PCSID) associated with QoS performance characteristics.	• Adopt the PCSID signalling
Extending the statistical service option through the hard service option	• Usage of routes learned via q-BGP (if q-BGP is activated by the AS offering statistical service option): if no indication is inserted in q-BGP messages, any solution option could pretend that this route is valid from a service point of view. As a result, this could generate black holes in the Internet.		• Adopt the double signalling channel. This could be achieved in at least two ways: • Duplicate the q-BGP announcements and indicate the service option it serves • As far as the hard solution option is considered, an AS will announce only its PCSs thanks to the use of an identifier (PCSID) associated with QoS performance characteristics.	• Adopt the PCSID signalling

2.6 Bidirectionality

The MESCAL solutions allow QoS-based IP delivery service between end-points spanning a substantial number of domains. The general requirements of providing bi-directional services with, possibly different, QoS assurances in the forward and reverse paths should be considered.

In the cascaded approach adopted by the project, each Network Provider (NP) or ISP forms *pSLS* contracts with adjacent NPs. Thus, the QoS peering agreements are only between BGP peers. This process is repeated recursively to provision QoS to reachable destinations that may be several domains away. Figure 10 shows an example for end-to-end uni-directional QoS service implementation using the cascaded approach. Each NP/ISP administers its own domain and the inter-connection links that it is responsible for. For example in Figure 10, ISP1 is responsible for the network provisioning and resource allocation in AS1 including the configuration of both *"**a**"* and *"**b**"* interfaces.

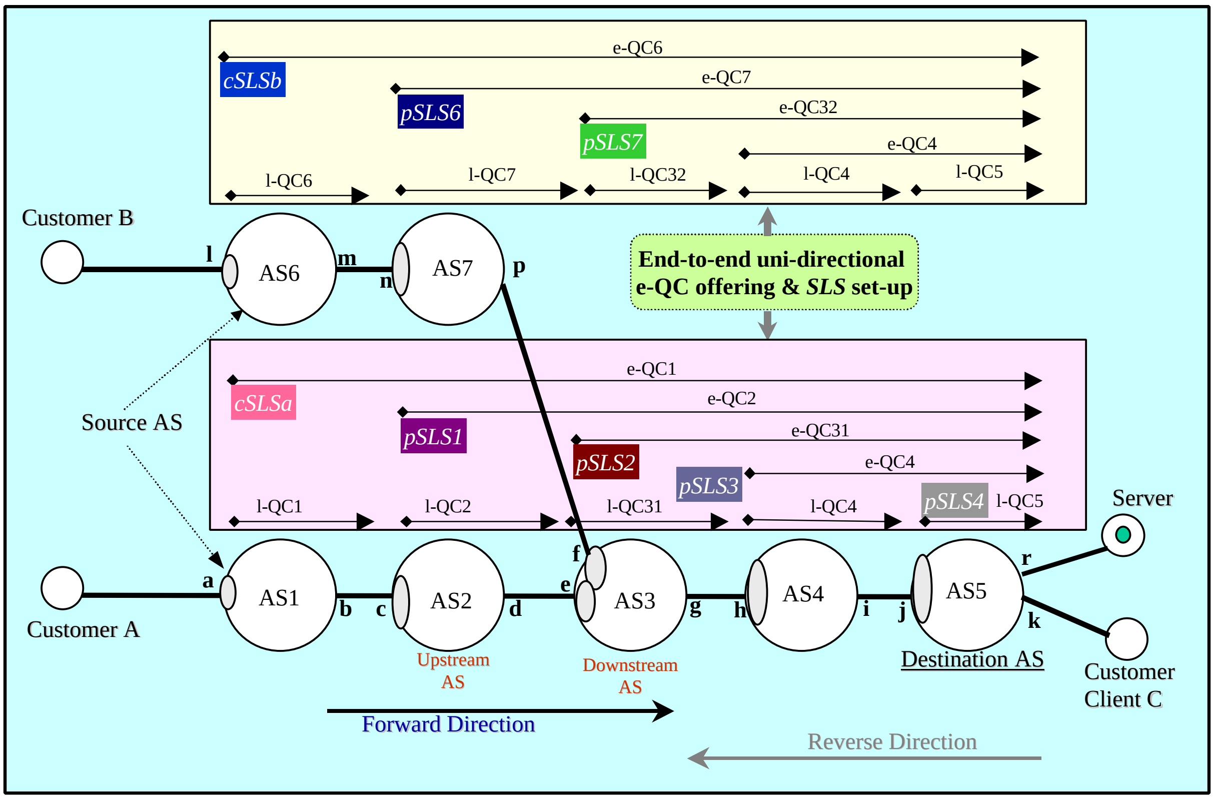


Figure 10. End-to-end uni-directional QoS service implementation

2.6.1 Bi-directionality in Statistical Guarantees Solution Option (2)

There are some fundamental problems to be solved in order provide bi-directional services with solution option 2. There are two methods to tackle the problem of providing QoS enabled path in reverse direction. The first method extends the single cascade with bi-directional capabilities. The second method employs a unidirectional cascade in each forward and reverse direction to build bi-directional services.

2.6.1.1 Method 1: Bi-directional pSLSs

One possible solution for setting up a reverse path is to negotiate *pSLSs* in reverse direction between peer ASs with an open destination scope (*). An open scope is necessary when considering that as the e-QC is sold on, it can become part of a new e-QC, the scope and QoS parameters of which cannot be known by the Destination AS. To allow the upstream AS to offer the e-QC to further upstream ASs without the need for amending the scope of pre-existing downstream pSLSs every time the scope changes, the (*) is required. This potentially solves the bi-directionality problem at the pSLS level, but it raises some issues in implementing the e-QCs and invoking the service as discussed further in deliverable D1.4 [D1.4].

Alternatively, bi-directionality could be tackled by employing e-QC enabled *c/pSLSs* in the forward direction and l-QC enabled *pSLSs* with no explicit e-QC binding in reverse direction. This could have scalability problems in some specific scenarios, however. For example it is possible that two different streams of return traffic originated from a destination to a source may use the same l-QC in one of the transit domains. This creates a splitting problem at the egress point of that domain. v-QCs can be used within the domain to differentiate the streams at the egress point of the domain but it implies that a v-QC is needed per p_rSLS unless additional state information is used for inspecting and classifying packets. Both implications raise scalability issues.

2.6.1.2 Method 2: Multiple unidirectional cascades

This method allows the establishment of uni-directional *SLSs* for sending traffic only. Bi-directionality is left to the application layer to resolve. The suitable e-QCs have to be set-up separately by the source and destination ASs. There is no guarantee that a suitable e-QC for the return path will exist for any given forward e-QC, except by virtue of a "customer God", who ensures that suitable reverse path e-QCs exist in the destination AS, based on application requirements. This method would potentially provide the environment for having bi-directional services using the cascaded approach in both directions.

In the case where a client wants to receive traffic from the server with a given QoS (e.g., to download a file), the client must contact the server at the application layer with a request to send traffic to the client. The QoS requirements of the sending traffic as well as the billing details are also agreed between the two. The application layer communication between customers or client/server will need a way to describe and agree on the QoS levels to be used in each direction. This could be done by exchanging details of the specific e-QCs they have subscribed to in their respective *cSLSs*, or it could be done at a more abstract level in a customer language without exposing exactly how this is mapped to the e-QCs/*cSLSs*/QoS parameters they have with their respective ISPs.

2.6.2 Bi-directionality in Loose Guarantees Solution Option (1)

In this solution option, an AS advertises the Meta-QoS-classes it supports within its administrative domain. Other domains can make pSLS arrangement with this domain to make use of offered Meta-QoS-classes . Thereafter, each domain can find out whether it can reach certain destinations in a Meta-QoS-class plane through qBGP updates it receives. *pSLSs* agreed between two domains are not tied with certain destinations as in solution option 2. Hence, as *pSLSs* are uni-directional and they are established for transporting traffic in forward direction, p_rSLS can be established for transporting traffic in reverse direction. The scopes for handling QoS of these two *pSLSs* are the same i.e., Meta-QoS-classes within the domain.

There might be a different Meta-QoS-class requirement in reverse direction than forward direction. To address this, there can be an application level communication between the two parties (customers) involved in order to specify the QoS requirements in either direction.

2.6.3 Bi-directionality in Hard Guarantees Solution Option (3)

Neighbouring domains establish pSLSs between themselves. q-BGP runs between the domains, which already have established pSLSs. Solution option three uses q-BGP to announce PCS unique identifiers across the Internet in order for "option-3" ASs to be able to discover a path towards every AS having a PCS. Therefore, when an AS wants to establish an LSP between 2 addresses, its PCS calculates a PCS-path towards the destination AS, and it is up to each AS in the PCS-path to establish the LSP. At the service/application level, when originating AS wants to establish an LSP to a destination ASs, there must be an agreement between the two ASs (PCSs). This agreement specifies both the tail-end address of the LSP, the PCS identifier of the destination AS and this is also used to verify the existence of service contract exists between the two. In order to have bi-directional communication, *pSLS* and p_rSLS can be set-up the same as solution option 1. Thus, based on these SLS, LSPs can be created in forward and reverse directions in order to build bi-directional services.

2.6.4 Conclusion

This Section has given an overview of the problems and solutions discussed in deliverable D1.4 [D1.4] for providing bi-directional services with the three MESCAL solution options. The main issue is how to construct the QoS-enabled reverse path for return traffic. For solution option 2 we believe the most feasible solution for providing bi-directionality is the use of multiple cascades for uni-directional e-QCs. Providing bi-directional services in solution options 1 and 3 causes less complication, because *pSLSs* are based on the Meta-QoS-class concept without specific end-to-end performance guarantees or predefined service scope in terms of reachable destinations.

A general conclusion for all solution option 1 and 2 is the requirement for service/application level signalling between the communicating parties. This is to find-out about the Meta-QoS-class plane for reverse direction, information for billing and admission control in solution option 1, to specify the desired sink for return traffic for the Destination AS and the l-QC/e-QC for return traffic, information for billing and admission control in solution option 2. In solution option 3, service level communication is also required to pass to source AS head-end of LSP and possibly PCSID of that domain and destination AS with tail-end of LSP and possibly PCSID of that domain and necessary information for authentication and billing purposes.

2.7 Network Provisioning Cycle

2.7.1 Network Planning and Provisioning

Network Planning is defined as the off-line processes that are responsible for determining the type, quantity and geographical location of the physical resources required by an IP Network Provider conduct its business by offering IP connectivity services to meet the predicted demand of its customers. According to the role of the IP Network Provider, as defined in the MESCAL business model, the physical resources in question include, points of presence, IP routers and the communications links interconnecting them, as well as mother equipment required for the operation of an IP network, such as management servers.

Network Provisioning is defined as the processes responsible for ensuring that the physical resources are deployed as planned and with the appropriate physical configuration. This is distinct from *Traffic Engineering,* which is responsible for managing the distribution of traffic, optimising the use of the deployed physical resources and ensuring QoS in a cost effective manner. In the MESCAL functional architecture TE is involved with the soft configuration of existing physical resources, which will be accomplished by setting and modifying OSPF weights, PHB bandwidth, qBGP route selection parameters as well as dynamically creating and updating the RIBs, FIBs etc.

Many network management activities, including traffic engineering, can be achieved automatically through configuring equipment via network management interfaces. The MESCAL solutions for SLS management and traffic engineering aim to deploy intelligent algorithms to meet this goal. On the other hand, the implementation of network planning decisions through network provisioning processes usually involves manual installation or configuration of physical equipment. This is clearly not something that can be automated, although it is possible to generate trouble tickets and work schedules this way. One aspect of network provisioning that could be achieved automatically, however, is the creation and modification of the transport capabilities of underlying physical networks to provide the required connectivity between the routers of the IP network.

The MESCAL business model assumes *Physical Connectivity Providers* (*Facilities Providers)* provide link layer pipes (e.g. electrical, optical, satellite) to interconnect the *IP Network Providers'* routers. Chapter 3 of deliverable 1.4 [1.4] considers the underlying transport network provided by Physical Connectivity Providers and how they may be interfaced to IP Network Providers offering one or more MESCAL service options.

Network provisioning can occur at a range of time scales. On a monthly scale new IP peering agreements will cause the network planner to request new or additional physical connectivity between IP Network peers. On a short time scale the Intra- and Inter- Domain provisioning cycles could cause the creation of new links and/or the modification of existing links' capacities via a management plane protocol, e.g. XML a la TEQUILA/MESCAL SrNP or control plane signalling, e.g. RSVP-TE, LCAS, (see D1.4).

2.7.2 Optical network technologies for dynamic network provisioning

The current most widespread approach to optical networking is the provisioning of static wavelengths within fibres in WDM (Wavelength Division Multiplexing) systems. Static point-to-point links provide fixed paths for wavelengths between two geographic locations. Network configuration is performed through manual configuration or via electrical switching. Electrical domain switching is not however fast enough for new applications and emerging line speeds and therefore new all-optical approaches to wavelength switching are being developed. To support the physical connectivity demands of MESCAL solution options at the fastest possible provisioning speeds with the least restrictions (capacity granularity, enforced topology, hierarchy etc.), it is envisioned that intelligent dynamic optical network would be required. While SDH and many other Layer 2 protocols could support the capacity requirement, their switching and transmission bandwidth limits are being approached. The emerging technologies considered in D1.4 are GMPLS (Generalised Multi-Protocol Label Switching) and ASON (Automatically Switched Optical Networks). These technologies provide an overlapping set of features that could be used in future networks to provide all optical dynamically re-configurable capacity.

GMPLS is the union of existing MPLS solutions, MPLambdaS (MP?S) and label switching through TDM networks. MPLambdaS provides for the configuration of optical forwarding as well as features associated with MPLS such as label nesting and link bundling. The only possible interoperability issue is that of capacity granularity and the ability to multiplex clients and sub-divide the bandwidth of each wavelength. Used together with link/LSP bundling in GMPLS or link aggregation in ASON, or VCat it would be possible to efficiently allocate fine-grained capacity up to very high speeds (multiple wavelengths).

The organisational separation of IP and optical networks would mean that there is no direct link between MESCAL Solution Options 1 and 2 and DWDM networks, and therefore any of the technologies considered in D1.4 would be suitable for the dynamic provisioning of bandwidth. MESCAL Solution Option 3’s use of MPLS however would allow for a closer integration as the IP network providers PCSs could now directly interface to the optical network’s PCSs for faster more efficient provisioning.

2.7.3 Network Provisioning in the MESCAL functional architecture

The hierarchical relationship between functional components and the development of the concept of “plan then take care” for the management and control of QoS in IP networks was developed in TEQUILA [TEQUI,D1.4] and adopted by MESCAL in D1.1. Network planning and provisioning fits into this hierarchy as follows:

Hierarchy of Management/Control functionality: “plan then take care”:

- Service Planning
  Defines services to be offered based on perceived customer demand and business objectives
- Network Planning/Provisioning
  Provisions sufficient physical network resources to meet service requirements
- Resource Provisioning/Traffic Engineering
  Configures the physical network, based on subscriptions
- Dynamic Traffic Engineering
  Dynamically adjusts network configuration based on actual traffic and network state (within limits imposed by off-line TE)
- Packet scheduling/forwarding
  Implements decisions of higher-layer algorithms in the data plane in real time

MESCAL has defined *Resource Provisioning Cycles*, both intra- and inter-domain to configure the network to meet perceived service demands (see Section 2.4 and Section 5). These currently assume that the physical network is fixed, although the TE functional blocks are assumed to raise alarms to the off-line network planning processes when they are unable to accommodate the traffic demands within the existing physical network by soft configuration alone.

Thanks to the emerging capabilities of modern optical networks it is now possible to conceive of network resources (link bandwidth) being provisioned dynamically, which could be exploited by a *Network Provisioning Cycle* within the MESCAL functional architecture. This would involve algorithms deployed within a *Network Planning* functional block, which may be invoked by *QoS-based Service Planning*, *Traffic Forecast*, or could be triggered by *Off-line Intra- or Inter-domain TE* when they are unable to satisfy the traffic demands within the existing resources.

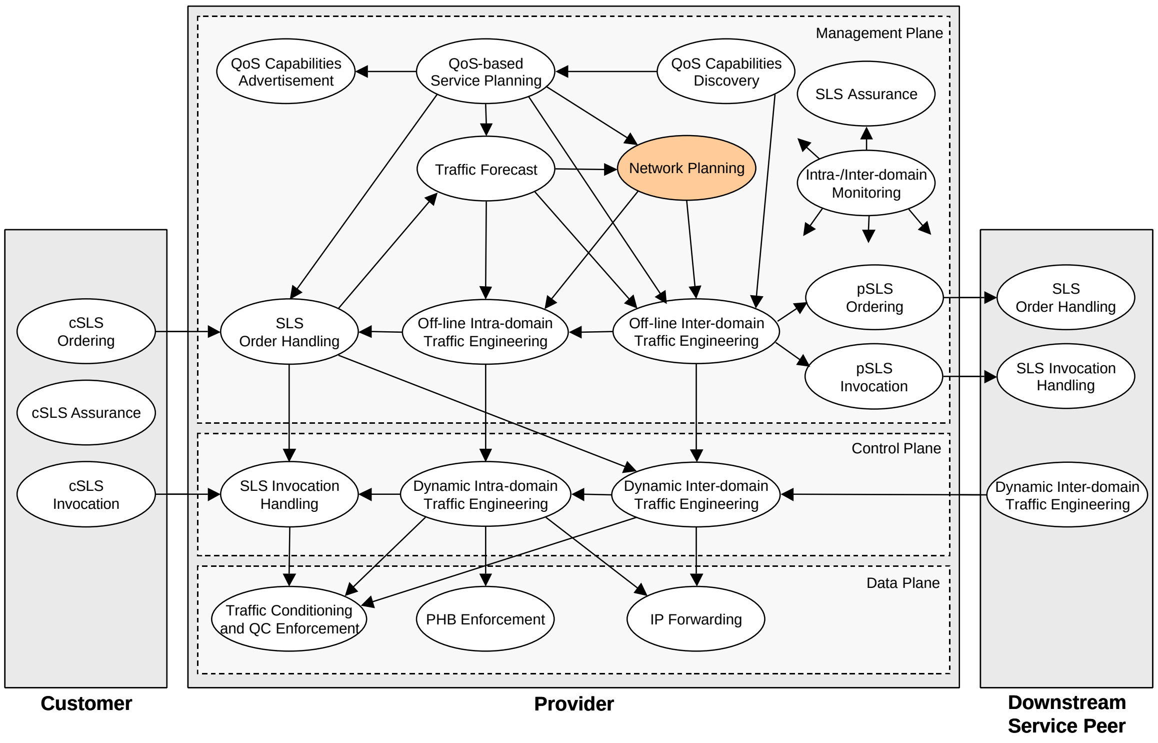

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Traffic Forecast
Network Planning
Intra-/Inter-domain Monitoring
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering
pSLS Ordering
pSLS Invocation
SLS Order Handling
SLS Invocation Handling
cSLS Assurance
Control Plane
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 11. Positioning of Network Planning in the MESCAL functional architecture

2.7.4 Network Planning interactions

Two scenarios are presented below for the triggering of the Network Provisioning Cycle: the creation of new service peers and the triggering of dynamic network provisioning by Off-line Intra-domain TE and Off-line Inter-domain TE during intra- and inter-domain Resource Provisioning Cycles.

2.7.4.1 Creation of new service peers

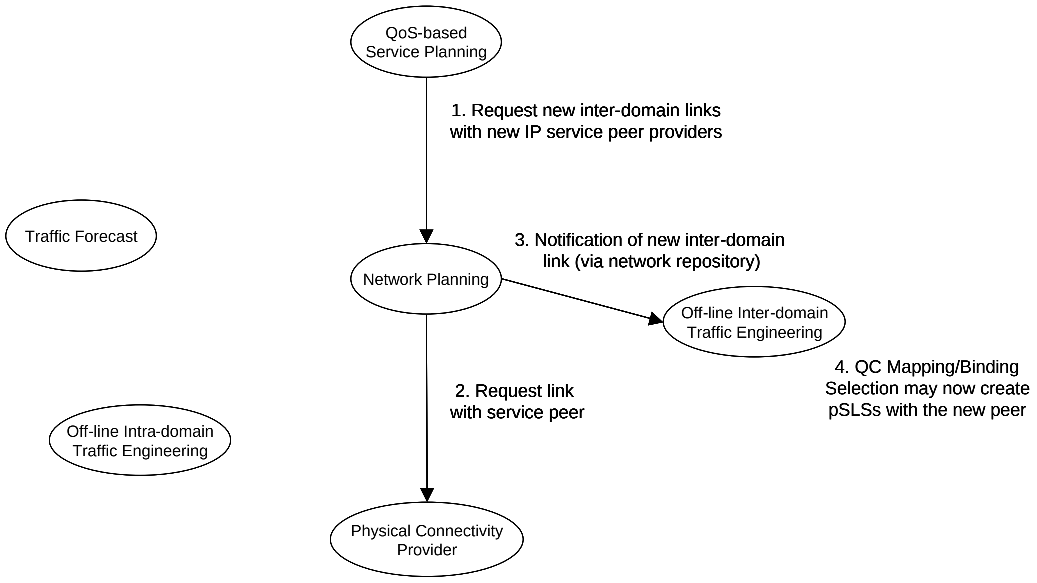

QoS-based Service Planning
1. Request new inter-domain links with new IP service peer providers
Traffic Forecast
Network Planning
3. Notification of new inter-domain link (via network repository)
Off-line Inter-domain Traffic Engineering
4. QC Mapping/Binding Selection may now create pSLSs with the new peer
2. Request link with service peer
Off-line Intra-domain Traffic Engineering
Physical Connectivity Provider

Figure 12. Interactions – creation of new service peers

Negotiation of peering agreements is assumed to be handled by QoS-based Service Planning and is completely out of the scope of MESCAL. Automatic provisioning of new links clearly implies that physical termination points are already in-place, and are suitably connected to the IP Network Provider's routers.

1. QoS-based Service Planning identifies an ISP with which it would like to peer and establish pSLSs in the future. This assumes that an appropriate business relationship has already been established with the new ISP, by QoS-based service planning or a higher layer function, probably manual.
2. Network Planning requests that a new link, e.g. optical wavelength within an existing fibre, is established with the new peer. This can be achieved via a management plane protocol, e.g. via web services.
3. Following the successful establishment of the link by the Physical Connectivity Provider Off-line Inter-domain TE is informed (probably via a network repository database).
4. pSLSs may now be negotiated with the new peer within the constraints of the capacity of the link established in step 2. Traffic demands for remote destinations may subsequently be engineered to use the new pSLS, through the standard inter-domain RPCs. Should it not be possible to accommodate the demands within the capacity of the links then the bandwidth may need to be modified by a further network provisioning cycle (see 2nd scenario, below).

2.7.4.2 Resource and Network Provisioning Cycles

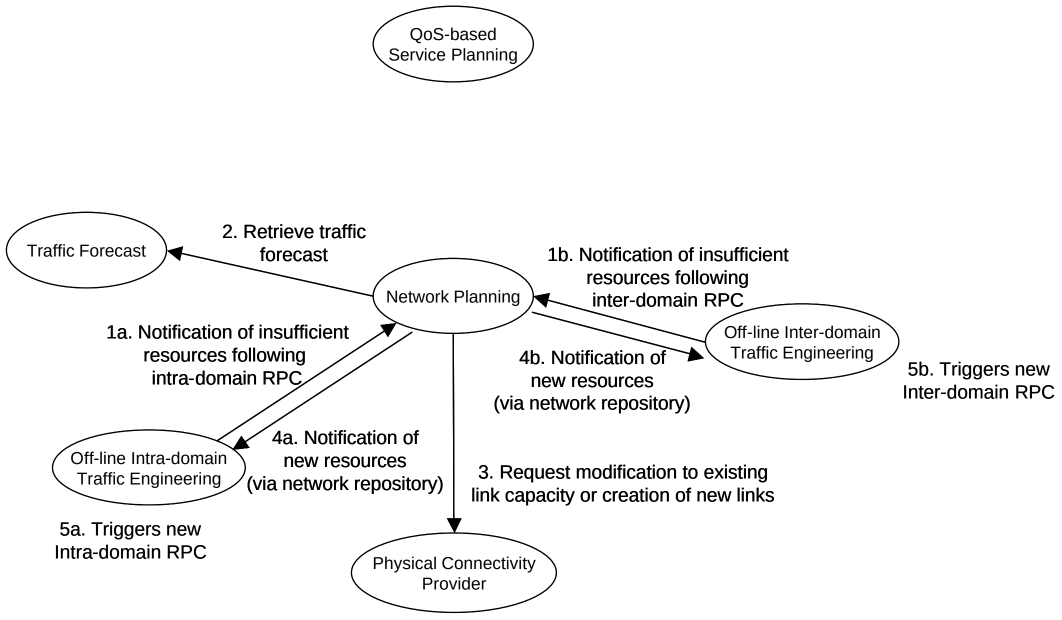

QoS-based Service Planning
Traffic Forecast
2. Retrieve traffic forecast
Network Planning
1b. Notification of insufficient resources following inter-domain RPC
Off-line Inter-domain Traffic Engineering
4b. Notification of new resources (via network repository)
5b. Triggers new Inter-domain RPC
1a. Notification of insufficient resources following intra-domain RPC
Off-line Intra-domain Traffic Engineering
4a. Notification of new resources (via network repository)
3. Request modification to existing link capacity or creation of new links
5a. Triggers new Intra-domain RPC
Physical Connectivity Provider

Figure 13. Interactions: Intra- and Inter-domain Network Provisioning Cycles

1. The intra- inter-domain TE algorithms identify that there is insufficient physical capacity to accommodate the forecasted traffic (or, that there is too much spare capacity, and therefore the network is operating inefficiently).
2. Network Planning retrieves a new traffic forecast (alternatively TF is aware of the lack of physical resources by the off-line TE components and this triggers Network Planning directly).
3. Network Planning applies its resource optimisation algorithms and determines the additional capacity to be provided by the Physical Connectivity Provider.
4. The new resources are made known to the off-line TE algorithms, via the network repository.
5. The off-line TE algorithms use the new physical resources in their future optimisations.

It should be noted that dynamic provisioning of link bandwidth assumes that the routers' interfaces can be appropriately configured for different rates. Furthermore, Network Planning needs to be aware of the granularity of bandwidth available from Physical Connectivity Provider and take this into account within its optimisation algorithms. This depends on technology – ATM, SDH, DWDM, GMPLS, etc.

2.7.5 Relationships between Network Planning and Traffic Engineering Algorithms

Traffic Engineering is assumed to operate within the constraints of the existing physical network. A common TE/planning algorithm for optimising physical and logical resources is not considered as this has major implications on the MESCAL TE algorithms and introduces too many degrees of freedom – not in the spirit of “plan then take care”. This is further justified by the fact that for the majority, if not all, operators, the underlying transport networks, such as SDH or DWDM, support a number of client networks, such as PSTN or leased lines, in addition to their IP network offerings. Furthermore the networks are often operated by different administrative divisions. A common TE policy across both client and server networks is unlikely to be deployed as the transport infrastructure has to be optimised for the demands made by all clients and not just the IP networks.

In a similar way to *Offline Inter-domain TE* interacting with *Off-line Intra-domain TE*, as described in Section 5, to achieve a loosely-coupled optimisation of both inter- and intra-domain resources, Network Planning may need to interact with intra- and inter-domain TE to investigate “what-if” scenarios before committing to buying new physical resources.

2.8 MESCAL multicast functional architecture

2.8.1 Overview

The proposed multicast functional architecture (shown in Figure 14) is consistent with the overall MESCAL scenario, and most of the components can be included in or mapped onto the blocks in the general architecture. In this way the corresponding implementation can be compatible with its unicast counterpart. From this point of view, the multicast architecture is not new nor is it independent of the general MESCAL model. For simplicity we do not include all the functional blocks in the overall architecture, but only illustrate the components that should be necessarily associated with multicast services. Meanwhile some new blocks are appended exclusively for multicast services (with * inside the block). On the other hand, there is one difference in defining service peers in the figure: we name the server side ISP (the right most part in the figure) the *upstream* provider instead of a downstream one because the multicast traffic is flowing in the opposite direction of the unicast flows. This implies that the domain level multicast SLS ordering/handling is always from the receiver to the source.

MSLS Order Handling is a subset of *SLS Order Handling* in the general architecture, and it is responsible for subscription level admission control on multicast customers. The most distinguishable aspect from its unicast counterpart is that the functional block negotiates with multicast group members/receivers instead of data sources. The Offline Multicast TE block will provide mSLS Order handling the resource availability of the engineered network for multicast traffic such that the later is able to decide whether to accept new mSLS requests for receiving multicast data. This type of mSLS requests can come from both local multicast customers and the ISP’s peering neighbours.

Offline Multicast TE can be further divided into intra- and inter-domain parts, which are respectively embedded in the corresponding offline TE blocks in the general MESCAL architecture. The task of this functional block is to map the demanded multicast flows onto the physical network resources and configure these resources in order to accommodate the forecasted traffic from both local customers and peering ISPs. Furthermore, in order to achieve end-to-end QoS requirements across domains, the QC mapping and binding selection/activation process still apply to the multicast scenario, and there should be minimum, if not no direct impact on the conventional mechanisms for unicast traffic. The process of Offline Multicast TE is also in a centralised manner within an AS during each RPC.

MpSLS Ordering is included in *pSLS Ordering* in the general architecture, and it interacts with the mSLS Order Handling block in the upstream service peer. Specifically, this block takes the responsibility of negotiating new multicast pSLSs with the upstream ISP, and this negotiation is based on the binding selection algorithms from the offline multicast TE block.

Dynamic Group Management can be appended to the *cSLS invocation handling* in the general architecture specifically for multicast services. In order to ensure that the network is not overwhelmed with multicast traffic resulted from the policy of over-reserving resources at the subscription level, admission control should be introduced in group management for rejecting excessive join requests on new group sessions. Moreover, this functional block should also have the capability of dealing with heterogeneous QoS requirements from members who subscribe to a common group session.

Similar to the offline scenario, *Dynamic multicast routing* can be regarded as part of the Dynamic TE blocks in the general architecture, and it has the functionality of constructing and updating real time multicast trees according to the group membership dynamics. When the Designated Router (DR) receives an IGMP membership report, the task is how to deliver the QoS join request towards the source, such that a feasible path can be found to carry the multicast traffic to the receiver. Moreover, this block should also provide capabilities of dynamic traffic engineering such as bandwidth conservation and load balancing etc.

mpSLS Invocation basically has the similar functionality to the corresponding *pSLS Invocation* in the general functional model. The only difference is that the interaction is with the upstream ISP in terms of the usage of multicast pSLS dynamics from receiver peer’s perspective.

PHB enforcement for multicast services is contained in its counterpart in the general architecture and it mainly considers how to treat multicast packets with proper PHBs at the core network. Compared to the unicast scenario, multicast packets can be replicated at any branching point where two or more join requests are merged together. How to treat replicated packets destined to group members with heterogeneous QoS requirements becomes a new issue. As it is known that conventional multicast trees are recorded through group state maintenance within the network, how to enable these trees to exhibit multiple PHBs without significantly extending core router forwarding architecture is another issue to be coped within this block.

Multicast forwarding is part of *IP forwarding* in the general architecture, and it basically has two tasks: first, when a multicast packet arrives at the incoming interface, the router should replicate it and forward the packets on all the outgoing interfaces where group join requests are received. Second, at each outgoing interface the replicated packets should be treated with proper PHBs that correspond to the original QCs expressed in the join requests from downstream group members. The behaviour of multicast forwarding should also obey the reverse path forwarding (RPF) rule.

RPF checking is a packet-level mechanism for avoiding loops dedicated to multicast traffic delivery. At each multicast router, if the packet is not received from the interface on the shortest path back to the source, this packet will be silently dropped. This guarantees that multicast traffic is always forwarded along the shortest path from the source to individual group members. In the MESCAL solutions, even if QoS routing is to be used in multicast tree construction, the multicast RPF checking mechanism should still take effects as a necessary constraint for multicast forwarding.

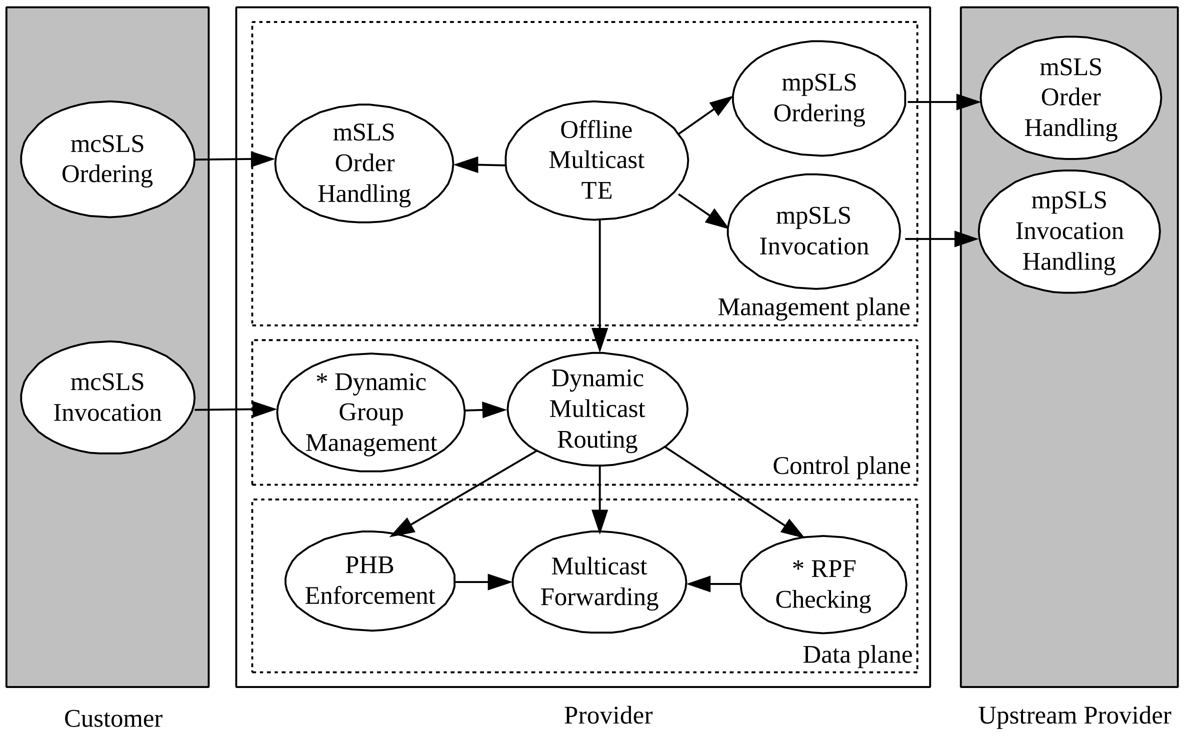


Figure 14. MESCAL multicast functional architecture

3 SERVICE PLANNING AND QOS CAPABILITIES EXCHANGE

3.1 Introduction

The *Service Planning and QoS Capabilities Exchange* functional group [D1.1, Section 6.1] describes the business decisions-related part of the Management plane of the MESCAL functional architecture. The functional block this group is divided to are highlighted in Figure 15 below. These two components are the *QoS based Service Planning* and the *QoS capabilities Advertisement and Discovery* functional blocks.

The planning component's purpose is to aid the AS's decision-making parties by providing them with informational statistics and planning analysis, and to automate the process of enforcing the business decisions, such as new service offerings, by triggering the relevant components of the MESCAL system. The advertisement and discovery components augment the functionality of the planning component by automating the process of QoS capabilities exchange between this AS and its potential business partners –i.e. other ASs.

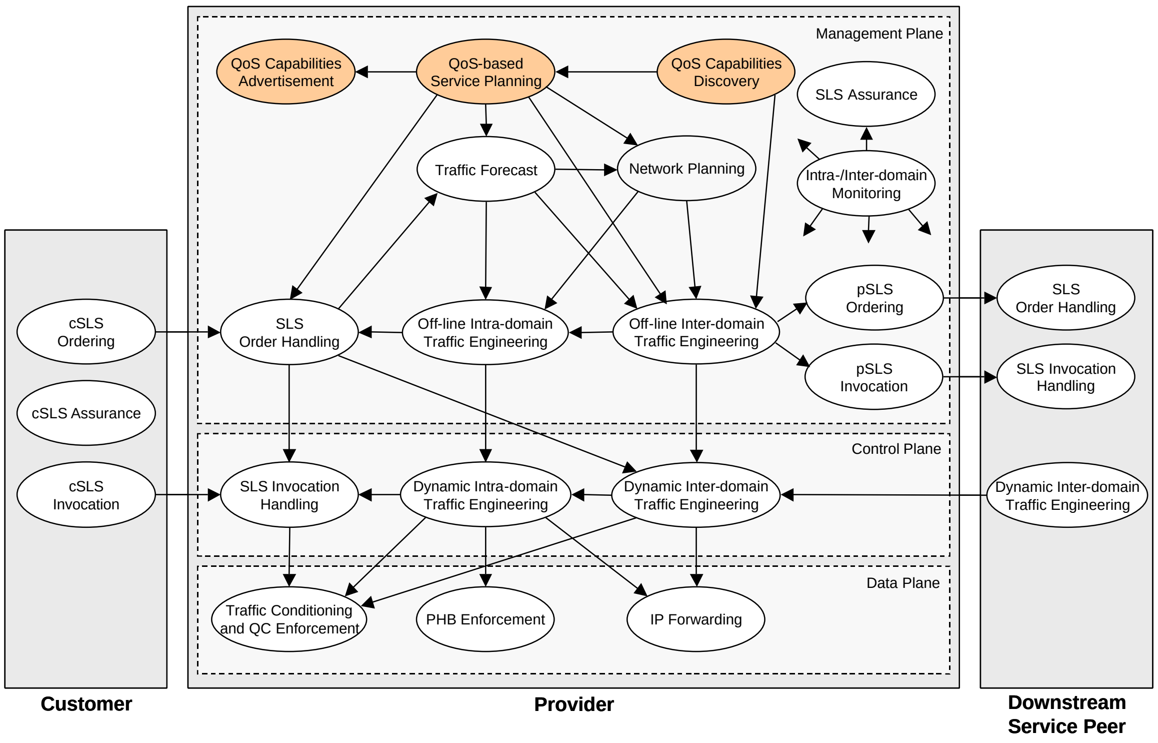

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Traffic Forecast
Network Planning
Intra-/Inter-domain Monitoring
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering
pSLS Ordering
SLS Order Handling
cSLS Assurance
pSLS Invocation
SLS Invocation Handling
Control Plane
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 15. Service Planning and QoS Capabilities Exchange

The interfaces –internal and external- of the Service Planning and QoS Capabilities Exchange functional group are specified in this Section. The scope of the functionality of each block will be analysed appropriately.

3.2 QoS-based Service Planning

3.2.1 Objectives

The main focus of the *QoS-Based Service Planning* functional block is the business decisions making processes. Its goal is to facilitate these processes by offering statistics and projections concerning the opportunities of QoS service offerings by the AS, and after a decision has been made to ensure its enforcement by the service provisioning mechanisms of the AS.

Figure 16 presents the QoS based Service Planning component together with the other functional components with which it interacts, within the same AS employing the MESCAL functional architecture. The behaviour of this component is influenced by marketing and business policies.

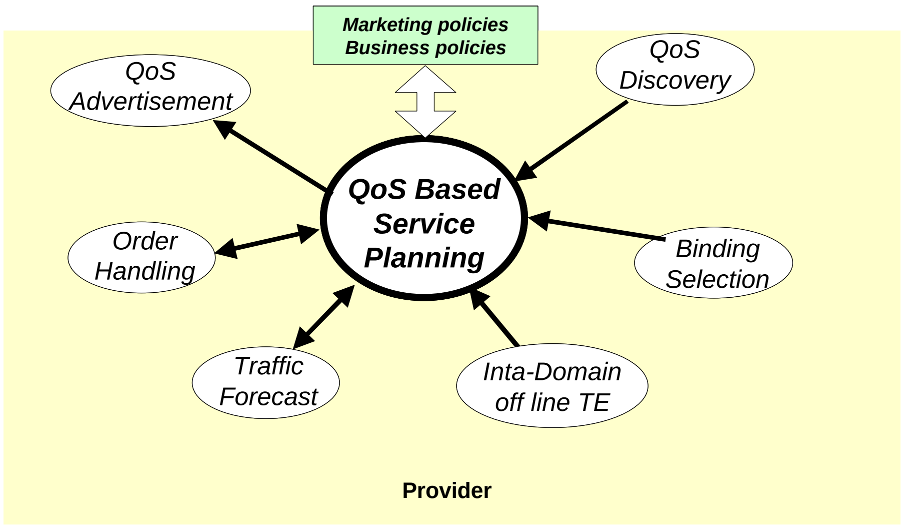


Figure 16. QoS-based Service Planning

3.2.2 Interface Specification

According to the MESCAL functional architecture *QoS-Based Service Planning* implements interfaces with several other components internal to the architecture, but no external interface:

- *QoS Capabilities Discovery*. Through this interface planning receives advertisements of pSLSs offerings by other ASs. These pSLS advertisements contain all the information inherent to pSLS technical aspects plus the terms and conditions for their purchase including cost. Not all advertisements that reach the discovery component are propagated to planning. Filters decided by planning and enforced by discovery will filter out irrelevant advertisements. In addition planning can use this interface to request an active search for pSLS offerings of specific attributes.
- *QoS Capabilities Advertisement*. Through this interface planning mandates the advertisement of the decided service offerings, pSLSs and cSLSs. The information passed to the *QoS Capabilities Advertisement* component is the service offering parameters, restrictions on these parameters, indicative costs and the desired advertisement campaigns.
- *SLS Order Handling*. Through this interface planning receives logs of conducted negotiations for the deduction of statistics concerning the requests of services issued to the AS and the negotiation outcomes. Planning also configures the *SLS Order Handling* function block so as to able to handle requests for the newly introduced QoS service offerings. This configuration includes the cSLS and pSLS templates, offering restrictions –in terms of SLS parameters–, admission logic policies and cost for each service. In the case of newly offered cSLSs the cSLS Offering web-Server is appropriately configured for handling the ordering of these new services by the end consumers.

- *Traffic Forecast*. Through this interface planning receives the currently valid traffic demand forecast factors per each offered service per each class differentiating usage behaviour. It also receives the established subscriptions. Planning uses this interface to configure traffic forecast algorithms with initial traffic demand predictions for the new services decided to be offered. These predictions will be refined by forecast mechanisms later on based on usage statistics.
- *Binding Selection*. Through this interface planning assists in the establishment of new pSLSs by expressing to *Binding Selection* the list of e-QCs that the domain wishes to offer, and giving to *Binding Selection* the set of l-QC capabilities of the AS, and setting the policies regulating this extension.
- *Offline Intra-Domain TE*. Planning receives the calculated resource availability matrix (RAM) each time a new cycle commences.
- Marketing / business logic. Through this interface planning informs the business decision making parties of the AS of statistics concerning service planning and of the results of its algorithms, analysing this data with the aim of deducing the optimum service offerings. Planning receives configuration policies influencing the outcome of its algorithms and approval of its decisions or direct orders, by business marketing authority, for the enforcement of specific service offerings.

3.2.3 Behaviour Specification

The functionality of the *QoS-based Service Planning* functional block aims to deliver the following results:

- Produce a set of potential pSLS and cSLS offerings including indicative costs, offering terms and restrictions as well as proposed advertisement policies. These results will be used by the marketing and business authorities that will make the final decision.
- Offer statistics deduced from data, collected by the operation of several components of the MESCAL architecture, concerning planning and aiming to facilitate the business authorities decisions.
- Realise the approved or ordered service offerings by appropriately configuring *SLS Order Handling*, *QoS Capabilities Advertisement* and *Traffic Forecast* function blocks.
- Decide on the potential e-QCs to be offered and policies influencing the binding decisions that will realise these e-QCs. The business authority must approve these decisions first.

3.3 QoS Capabilities Discovery and Advertisement

3.3.1 Objectives

The objectives of the QoS Capabilities Discovery component are to inform the planning component of the advertised capabilities of other ASs by filtering received messages or by performing search for requested QoS connectivity capabilities, initiated by planning queries.

The objectives of the QoS Capabilities Advertisement component is to advertise the QoS connectivity services offered by this provider, as decided by planning, to other ASs and to end consumers.

Figure 17 presents the *QoS Capabilities Advertisement and Discovery* function block of a provider along with its interactions with the internal functional components of the MESCAL architecture and the interactions with external entities such as a virtual market for QoS connectivity and other ASs.

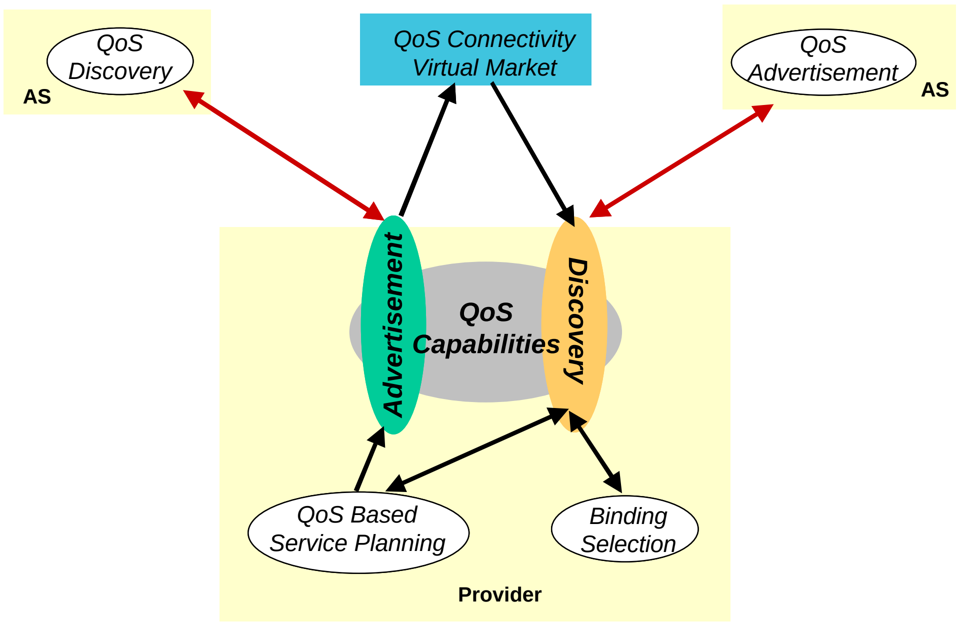


Figure 17. Advertisement and Discovery

3.3.2 Interface Specification

QoS Capabilities Discovery and Advertisement functionality can be divided to the advertisement related and the discovery related process. Each process implements external interfaces for communicating with other ASs or third parties acting as brokers and internal interfaces to exchange information with the planning and the binding components as defined by the MESCAL functional architecture.

3.3.2.1 External Interface

- QoS connectivity Virtual Markets.

The Advertisement process implements an interface that allows it to publicise advertisements of the provider's offered QoS connectivity services. These advertisements can be customised and targeted to specific consumer groups, also they can be passive, discovered by consumers search, or active, shipped to the consumers.

The Discovery process implements an interface that allows it to subscribe in order to receive selectively -by setting the desired filters- advertisements of QoS connectivity services offered by other providers. Through this interface it could also launch searches for service offerings fulfilling certain desired criteria or even publicise open requests for specific services.

- Known ASs.

The Advertisement process implements an interface that allows it to communicate directly with the QoS discovery component of other ASs. These ASs maintain relationships with this provider so their contact point is well known. Through this interface appropriate advertisements according to the relationship between the provider and each AS can be shipped.

The Discovery process implements an interface that allows it to communicate directly with the QoS advertisement component of other ASs. Same case as before, the ASs maintain relationships with this provider. Through this interface the provider receives advertisements within the limitations imposed by the relationships between the ASs, and also the provider can submit queries for desired services.

3.3.2.2 Internal Interface

- *QoS-based Service Planning*. Through this interface QoS advertisement and discovery receives this AS' service offerings and their related advertisement policies. Also the interest of planning for specific service offers is expressed, to be translated to advertisement filters by discovery process. In addition requests for active searches for service offerings of specific attributes may be received. Service offerings acquired on demand or filtered from the received advertisements are forwarded to planning through this interface
- *Binding Selection*. Through this interface *Binding Selection* requests and receives the available QoS connectivity offerings –in terms of offered pSLSs (in particular, the o-QC and destination address prefixes)- that fulfil certain desired criteria.

3.3.3 Behaviour Specification

The functionality of the *QoS Capabilities Advertisement and Discovery* function block aims to deliver the following results:

- Advertisement of the service offerings of the provider through publication to relative virtual markets. This publication is done under specific terms dictating the targeted consumers and the advertisement methods employed. A suitable technology for the implementation of such functionality is the emerging technology of web services including WSDL base language for defining advertisements and UDDI protocol for realising the communications.
- Discovery of available service offerings from other ASs, satisfying certain criteria though virtual markets. The discovery involves the subscription for receiving desired advertisements, the active search for fitting service offerings and the publication of service requests. The implementation of this functionality could be based on web services, as mentioned before, since both advertisement and discovery of services through virtual marketplaces is an undivided system.
- Advertisements of the service offerings of the provider directly to other providers having a relationship with him. The advertisement process must maintain a list containing all the known ASs, their contact points and their relationship with this provider which determines the relative advertisement policies. Each new service offering, direct advertisement decided by planning can be realised by using this data. A suitable technology for implementing this direct communication is the soap protocol –also employed by web services- and for implementing the list of contacts is current database technology.
- Discover of available service offerings from other ASs directly. This includes the direct reception of advertisement and their filtering, based on criteria dictated by the needs of planning, and the direct querying of other ASs for their offered services. This functionality is realised using the maintained list of ASs as before and could be implemented using the same technologies.

4 SLS MANAGEMENT

4.1 Introduction

The SLS Management functionality is a major part of the management plane of the MESCAL functional architecture as introduced by deliverable D1.1 [D1.1]. By SLS Management we name the functionality responsible for handling the technical contracts -modelled as cSLS and pSLS- that specify the QoS connectivity services offered or acquired by the provider. SLS Management is essential for providing QoS connectivity services because it undertakes the task of establishing agreements that will allow the provider to expand its network's QoS connectivity beyond its domain, as well as the task of responding to the requests for services from the provider's customers taking into account predictions of the capacity of the network and business directives.

The SLS Management functionality can be split into two parts: (a) the part responsible for the contracts offered by the provider to its customers, i.e. the end-customers and interconnected providers, and (b) the part responsible for the contracts requested by the provider from its peer providers. The resulting functional components are named "*SLS Order Handling*" and "*SLS Ordering*" respectively. The communication between these components is based on a protocol especially designed for the conduction of the required negotiations for service purchase, the SrNP (Service Negotiation) protocol. While the ordering process establishes the contracts between the peering providers, the invocation process is required to commit resources before traffic can be exchanged, with "*SLS Invocation Handling*" and "*pSLS Invocation*" providing the necessary functionality.

The MESCAL functional architecture is presented in Figure 18 below with the components realising the SLS Management functionality highlighted.

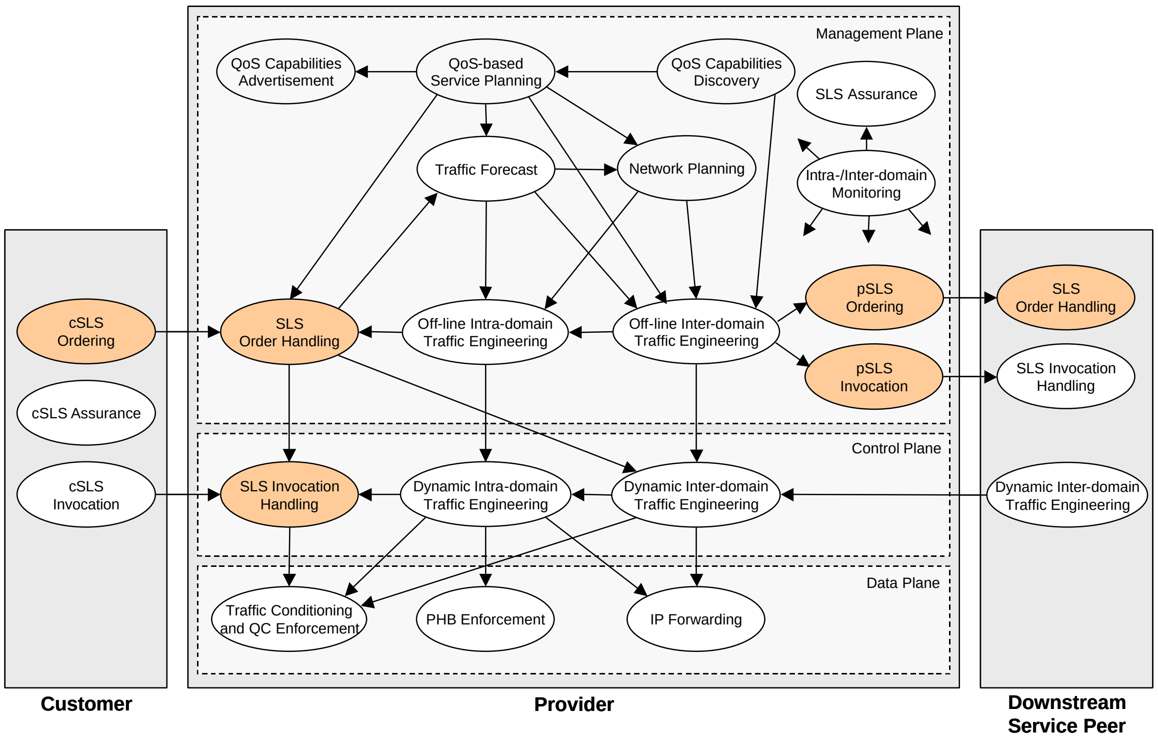

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Traffic Forecast
Network Planning
Intra-/Inter-domain Monitoring
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering
pSLS Ordering
pSLS Invocation
SLS Order Handling
SLS Invocation Handling
cSLS Assurance
Control Plane
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 18. The MESCAL functional architecture

This Section contains the following contents. Specifications for pSLSs and cSLSs are given in Section 4.2. The Service Negotiation Protocol (SrNP) for inter-domain QoS is described in Section 4.3. The MESCAL functional architecture function blocks are then detailed: *SLS Order Handling* in Section 4.4, *pSLS Ordering* in Section 4.5, *cSLS Ordering* in Section 4.6, *pSLS Invocation* in Section 4.7, and finally pSLS *Invocation Handling* in Section 4.8.

4.2 cSLS and pSLS specifications

4.2.1 Introduction

The current trend in service offering is agreement (contract)-based. The term *Service Level Agreement (SLA)* is widely used to denote such an agreement. It describes the characteristics of the service offering and the mutual responsibilities of the parties involved for using/providing the offered service. The term *Service Level Specifications (SLS)* is used to denote the technical characteristics of the service offered in the context of an SLA. The service technical characteristics refer to the provisioning aspects of the service e.g. request, activation and delivery aspects from network perspectives. Non-technical service provisioning aspects such as billing and payment aspects, are not part of the SLS; they are part of the overall SLA. SLS are an integral part of a SLA, and conversely a SLA includes SLS.

MESCAL is concerned with SLS; service accounting and billing aspects are outside the scope of investigation. As the MESCAL solution for QoS delivery in the Internet adopts a hop-by-hp, cascaded model of interactions between providers, both at the service and network (IP) layer, we distinguish two types of SLS (and subsequently of SLAs):

- *cSLS*, established between end-customers and providers, and
- *pSLS*, established between providers with the purpose to back-up agreements at a service level for expanding the geographical span of their services.

The definition of c/pSLS, from an informational viewpoint, is the main theme of this chapter. Specifically, this chapter specifies suitable templates, set of parameters with clear semantics, for completely describing the contents of c/pSLS. We firmly believe that there is a need for standardising c/pSLS to the benefit of Internet service deployment and provisioning. Standardised c/pSLS would provide a common informational basis for the interactions between end-customers and providers and between providers, as well as for building the required service provisioning functionality; thus, enabling the automation of the respective processes.

The essence of our specification work is to look at pSLS under two angles: (a) as agreements between providers for QoS-traffic exchange, pertinent to the particular relationships holding in the business model for QoS provisioning in the Internet and (b) as QoS-based services offered by a provider. Clearly, there is a strong interrelation between these two aspects and each poses its own requirements on the content of pSLS. Analysing these requirements, suitable templates are specified. Note that this does not apply to cSLS, which mainly encompass QoS-based service offering aspects. As such, without loss of generality, we focus on pSLS; cSLS are quite similar.

The Section is organised as follows. First, the requirements underlying our specification work are outlined along line the different types of pSLS, which can be distinguished depending on the business context they are to be established in. Subsequently, by viewing that c/pSLS represent the QoS-based connectivity services offered by providers, templates for describing in detail all aspects of QoS-based services are specified. While these templates present an open, detailed pSLS model, suitable condensed, summarised pSLS models are then specified, as appropriate as required by the specific requirements posed by the different types of business relationships between providers.

4.2.2 Types of pSLS and Specification Requirements

This Section is specific to pSLS, not to cSLS.

pSLS form the basis of the agreements between providers for traffic exchange in the Internet. In essence, pSLS extend, to the end of QoS traffic exchange, the respective agreements that exist today between the providers in the best-effort Internet -for transiting or inter-exchanging traffic- [HUST]. As such, they should be in-line with the specific context of traffic exchange, which is implied by the particular business relationships holding between providers.

The business model, relationships and financial settlements between providers in today's best-effort Internet as well as in the MESCAL-enabled QoS-aware Internet are described and discussed from QoS perspectives in MESCAL deliverable D1.4 [D1.4, chapter 6]. Based on the analysis therein, the MESCAL solution advocates two basic business cases:

- A business case for the provisioning of QoS-based services relying only on loose QoS guarantees - qualitatively expressed performance targets, and no bandwidth guarantees.
- A business case for the provisioning of QoS-based services relying on statistical guarantees for quantitative performance targets and bandwidth, in addition to qualitative QoS guarantees.

It should be noted that in either of the above business cases, services relying on hard QoS guarantees could also be provided, by establishing MPLS-based tunnels (LSPs) between specific points in the Internet; however, this service is not for the mass market because of scalability limitations inherent in the technical solution.

The qualitative-QoS Internet business case directly corresponds to the three-tier, *hierarchical* model currently in place in the best-effort Internet. In this case, the business relationships between providers are completely determined by their relative positioning in the hierarchy; the following types are distinguished:

- *The MESCAL pSLS-based customer-provider* relationship, whereby, one provider -said to be acting in a provider-business-role- provides the QoS Internet connectivity service, as seen by its domain, to the other provider -said to be acting in a customer-business-role. Usually, this type of business relationship is between providers belonging to different levels of the three-tier Internet model, with the provider in the lower tier being a customer of the provider in the upper tier.
- *The MESCAL pSLS-based peer-to-peer* relationship, whereby, the providers mutually agree to exchange QoS traffic between their domains; not transiting traffic to their providers or to other peer-to-peer providers, although the latter could be a possibility. This relationship is a kind of 'short-cut' to prevent traffic flowing into the upper tiers and, usually, is between providers of similar size -belonging to the same tier.

The statistical-QoS Internet business case advocates a *flat* Internet, where the business relationships between providers are not affected nor dictated by the relative positioning of the providers in the three-tier hierarchy; we propose the following common type of business relationships between providers:

- *The MESCAL pSLS-based (upstream)-QoS-proxy* relationship, whereby, either of the providers may request from the other provider to provide a transit QoS-based connectivity service to (a subset of) anywhere the latter provider can reach in the Internet with this QoS. The provider offering the transit QoS service would have built its QoS reach capabilities based on similar agreements with (some of) its directly attached providers, which in turn would have built their own QoS reach capabilities based on similar agreements with (some of) their own adjacencies and so on. Therefore, each provider in a chain of QoS-proxy relationships established in the same direction appears as kind of a 'proxy' of the providers further along this direction. This type of business relationship is of a strong transitive nature, while is not following a strict customer-provider business paradigm; it could be thought as being the QoS Internet counterpart of a call-termination agreement in the PSTN and VoIP business world.

Based on the above discussion, the following different types of pSLS are distinguished:

In the hierarchical Internet business case:

- *pSLS for Internet access at loose QoS* –suitable for customer-provider business relationships, offered by providers wishing to undertake a provider-business-role
- *pSLS for loose QoS tunnels in the Internet* –as above
- *pSLS for traffic inter-exchange at a loose QoS* –suitable for providers wishing to establish corresponding peer-to-peer business relationships
- *pSLS for loose QoS tunnel extension* (the term extension is meant from/to this domain to/from another domain) –as above

In the flat Internet business case:

- *pSLS for Internet access at a statistically guaranteed QoS* - suitable for QoS-proxy business relationships, offered by providers wishing to provide a QoS transit service
- *pSLS for statistically guaranteed QoS tunnels in the Internet* –as above

The pSLS identified above differ each other in the type of the offered QoS guarantees, the directionality and topological scope of the traffic flows, according to the business case and the particular context of business relationship they refer.

In customer-provider business relationship, pSLS have the connotation of agreements for the provider in the customer-business-role to 'join in' (send and receive traffic) the QoS-aware Internet as seen by the other provider. They imply a bi-directional flow of QoS traffic and can only offer qualitative QoS guarantees to all destinations that can be reached by the provider in the provider-business-role.

pSLSs between peer-to-peer providers have the connotation of mutual agreements for the exchange of QoS traffic from one provider domain to the other provider domain. They imply a bi-directional flow of QoS traffic and offer qualitative QoS guarantees within the scope of the provider domains.

In the upstream-QoS-proxy business relationship, pSLS have the connotation of agreements for the provider offering the pSLS, pSLS-provider, to deliver QoS traffic from the other provider, pSLS-requestor, to (a subset of) the destinations that can be reached from the pSLS-provider with this QoS. They imply a unidirectional flow of QoS traffic, from the pSLS-requestor to the pSLS-provider and may offer statistical and/or qualitative QoS guarantees to certain destinations in the Internet -those reachable by the pSLS-provider.

Once pSLS are in place allowing providers to establish inter-domain QoS tunnels, these providers could offer to their end-customers cSLS with hard QoS and bandwidth guarantees.

Up to now, by viewing pSLS as agreements underlying the business relationships between providers, we analysed their intrinsic aspects regarding the characteristics of the QoS traffic flows that they imply. A number of different types of pSLS are required, as a result of the different types of business relationships that may hold between providers in a MESCAL-enabled QoS-aware Internet. Subsequently, the pSLS information specification task has to meet the following challenge:

- Provide for a common, ‘well-known and understood’ vocabulary to describe pSLS contents in a way that can satisfactorily fulfil the following two diverse requirements:
    - capture the essential aspects of the agreements between providers for QoS traffic exchange as implied by their business relationships, to the benefit of facilitating provider interactions and therefore service deployment in the Internet–different pSLS are bound to exist; while at the same time
    - considering that pSLS express QoS-based service offers, create a stable informational basis for building service management and traffic engineering functions, to the benefit of automated service provisioning and graceful delivery; although there may be a number of different pSLS, they should be supported by a common set of functions.

To the above end, MESCAL first specifies a general, open, detailed service model for describing pSLS as well as any QoS-based service; thus fulfilling the latter requirement. Subsequently, by appropriately restricting and/or summarising the information identified in this open service model, suitable models for describing the different types of pSLS, as identified per business case, are specified; thus fulfilling the former requirement. It should be noted that our specifications concentrate on service connectivity aspects; aspects such as service accounting, monitoring, billing and payment are not included. All these are presented in the subsequent sections.

4.2.3 A general model for pSLS and QoS-based services

By viewing pSLS as QoS-based service offers, this section specifies a general model for describing the technical (connectivity) aspects of such services, which are required for their provisioning and need to be agreed upon the provider and its end-customers or its peering providers i.e. their SLS.

Our work draws from the SLS template specification work of the IST TEQUILA project [TEQUI]. TEQUILA specified a service management and traffic engineering framework for intra-domain QoS provisioning [GOD02A], [TEQUI,D1.4], [GODER02], which prompts for standardisation of the notion of SLS, proposing a standard template. The proposed *SLS template (SLS-T)* is considered as the nucleus of IP QoS-based services. Broadly speaking, recognising that from connectivity perspectives, QoS-based services may be comprised of several ‘connectivity legs’ (e.g. between several sites), SLS-T describes the technical characteristics of a single ‘connectivity leg’ -topology, IP flows, transfer quality characteristics, traffic compliance criteria. The connectivity aspects of a service then, are the collection of suitable SLS-T’s bound to the same customer and the same access/usage means and characteristics. As such, the TEQUILA service management framework has specified the notion of *SSS-T (Service Subscription Structure Template)*, which may contain a number SLS-Ts, to describe the whole of the connectivity aspects of a QoS-aware IP connectivity service.

The SSS-T is the general, open, detailed model for pSLS and in general for SLS of any QoS-based service, adopted by MESCAL. The particular instances of the SLS-T and SSS-T templates for a particular QoS-based service are simply denoted by SLS and SSS. Note that in the context of specifications, the term SLS is meant as a unidirectional connectivity leg of a QoS-aware service, whereas, in any other context, this term denotes the general technical characteristics of a QoS-aware service; the latter corresponds to a SSS in the context of specifications.

The following sections present the TEQUILA-based SLS and SSS templates, highlighting the enhancements and clarifications that need to be made according to the MESCAL inter-domain perspectives.

4.2.3.1 SLS-T Specifications

SLS-T is specified against the following information elements (clauses), which are described in the following:

- SLS Identification
- Scope
- Flow Identification
- Traffic Conformance (Envelope)
- Excess Treatment
- Performance Guarantees

4.2.3.1.1 SLS Identification

A key, uniquely identifying the SLS in the context of a SSS; it is set by the provider.

4.2.3.1.2 Scope

The *Scope* clause explicitly identifies the geographical/topological region over which the QoS policy, as specified by this SLS, is to be enforced by indicating the boundaries of that region. It includes the following attributes:

- *Ingress,* indicating the entry point of the region over which SLS is to hold
- *Egress,* indicating the exit point of the region over which SLS if to hold

The *Ingress* and *Egress* attributes can take the following values:

<interface identifier | set of interface identifiers | label | any>, where:

"|" denotes an exclusive OR, "label" denotes a mutually agreed upon identifier uniquely identifying a boundary link and "any" is logically equivalent to unspecified.

The *Ingress/Egress* interface identifier may be an IP address or a layer-two identifier in case of Ethernet or unnumbered PPP-based access links in case of Point-to-Point Protocol or any other well-defined identifier uniquely determining a boundary link as defined in [RFC-2475].

The definition of *Ingress/Egress* is necessitated by the fact that providers cannot provide for QoS guarantees over the aggregation/distributions networks that usually intermediate between end-customers and provider domains. The values of the *Ingress/Egress* attributes may be deduced by other SLS or SSS attributes or by the traffic engineering functions of the provider. Usually, in the case of inter-domain services offered to end-customers and agreements with peering providers, one of these attributes corresponds to the interface of a customer access or an interconnection link, while the other attribute is left unspecified or set to an appropriate label denoting at high-level the points where liability for QoS policy enforcement ends for reaching a specific set of destinations or the boundaries of a particular domain (ASs). As an example, in the case of Internet access services offered to end-customers the value of the *Ingress* of the upstream SLS could be deduced by the customer information and the value of the *Egress* would be "any"; the latter could be refined to denote the interface of a particular inter-domain link by the traffic engineering functions, however, this is an internal matter, being not subject of agreement. In the case of VPN services offered to end-customers, both these attributes should be clearly specified.

The *Ingress/Egress* should not be confused with the characteristics of the flows entitled to receive the treatment of this SLS (cf. *Flow Identification* clause, below). They are quite distinct in semantics. Ingress/Egress, if specified, imply that the SLS traffic will have to pass through these points (interfaces); the issue is discussed in more detail in section 4.2.3.1.3.

The following combinations of Ingress, Egress values are allowed:

(1,1) - implying an one-to-one communication; we call the SLS as *a pipe SLS*

(1,N) - one-to-many communication (N>1); we call the SLS as a *hose SLS*

(1,any) - one-to-any communication; we call the SLS as an *unspecified hose SLS*

(N,1) - many-to-one communication (N>1); we call the SLS as a funnel SLS

(any,1) - any-to-one communication; we call the SLS as an *unspecified funnel SLS*

Because SLS in this template are assumed unidirectional QoS-based connectivity (legs of) services, the above taxonomy excludes the many-to-many communication (M, N); either *Ingress* or *Egress* attributes must be specified to exactly one interface identifier. Many-to-many communication can be achieved at the level of SSS, where a number of SLS are combined.

4.2.3.1.3 Flow Identification

The *Flow Identification* (*Flow Id*) clause defines the stream of IP datagrams, at an IP level, for which, the QoS policy, as specified by this SLS, is to be enforced. It includes the following attributes:

- *Differentiated Services Information*, specifying possible values of the DSCP field in the IP header for characterising the packets entitled to the SLS; it can take the following values: <DSCP value | set of DSCP values | any>
- *Source Information*, specifying possible values of the source IP address field in the IP header for characterising the packets entitled to the SLS; it can take the following values: <source IP address | set of source IP addresses | source IP prefix | set of source IP prefixes | any>
- *Destination Information*, specifying possible values of the destination IP address field in the IP header for characterising the packets entitled to the SLS; it can take the following values: <destination IP address | set of destination IP addresses | destination IP prefix | set of destination IP prefixes | any>
- *Application Information*, specifying possible values of application-related fields in the IP header for characterising the packets entitled to the SLS; it can take the following values: <combinations of sets of protocol number, source port, destination port | any>

The term "any" appearing above is logically equivalent to all.

Usually, each SLS must always have a single *Flow Id* clause with specified information along the above attributes. This is dependent on the nature of service; for instance, this clause may not be specified in the case of services for QoS tunnel set-up.

In essence, the *Flow Id* clause provides the necessary information for classifying the packets at the provider inbound link (cf. *Ingress*). The necessary information is included to enforce either an Aggregate (BA)- or a Multi-Field (MF)-based classification. In case of MF-classification, all above attributes may be specified; this classification may depict micro-flows as well as aggregate macro-flows. In case of BA-classification, the *Differentiated Services Information* attribute i.e. DSCP information must be specified, while the other attributes must not be specified. For scalability and performance reasons, especially for inter-domain services and related agreements, a BA-based classification is highly recommended; one should avoid fine-grained classifications or classifications based on multiple fields, even though aggregate traffic.

It should noted that the DSCP-value(s) specified in this clause, has(have) as such nothing to do with the DSCP-marking of packets inside the domain. The information included in this clause is solely to the purpose of identifying the traffic belonging to the contract underlining the SLS; therefore is agreement-specific, not behaviour/engineered-capabilities' specific nor revealing. Following classification, packets may be remarked by the provider to the appropriate DSCP, as required to receive the QoS treatment specified by the SLS. In the case of inter-domain services, the packets when leaving the domain may need to be remarked again to the DSCP corresponding to the SLS established with the determined next-hop peering provider, where they may need to be remarked again to the appropriate domain-specific DSCP to receive the required QoS treatment and so on.

Finally, the relationship between the *Scope* and *Flow Id* SLS information and their implications to routing are discussed. In general, if only *Flow Id* is specified and the *Ingress/Egress* are unspecified, or specified at a high-level by means of a label denoting the boundaries of a domain where QoS enforcement liability ends for reaching specific destinations, then, this is taken that there is no a-priori assumption about the actual *Ingress/Egress* points that the traffic will cross. Indeed, it is the responsibility of the provider to define the most appropriate route through its intra and inter-domain traffic engineering and routing policies. Thus, in this case, the *Ingress/Egress* information, which in this case is not an explicit part of the SLS, is implicitly derived by the routing policy of the provider. On the other hand, if both *Flow Id* and *Ingress/Egress* are explicitly specified, say by the pairs (DSCP, IP-src, IP-dest) and (IP-ingr, IP-egr) respectively, then, it is taken, that IP packets, adhering to the *Flow Id* information, must follow the route (IP-src, ..., IP-ingr, ..., IP-egr, ..., IP-dest). Conclusively, the information under the *Scope* and *Flow Id* clauses has different semantics, although in some cases unspecified information in one clause could be implicitly derived by the specified information in the other clause. Further, when information in both clauses is specified, this poses requirements on routing in that: the specified *Ingress/Egress* in the *Scope* clause should always be en-route of the packets specified in the *Flow Id* clause, in other words, packets must always be routed through the *Ingress/Egress* points, if these are specified.

4.2.3.1.4 Traffic Conformance (Envelope)

The *Traffic Conformance* clause describes the criteria (characteristics) that the traffic injected in the provider domain should comply with, in order to get the QoS guarantees specified by the *Performance Guarantees* clause. In essence, this clause sets the sufficient conditions at a traffic-rate level, that is, for the flows of the packets, entitled to the SLS (cf. Flow Id clause), to receive the specified QoS. It includes the following attributes:

- *Traffic Conformance Algorithm*, specifying the type of the mechanism, which is used to unambiguously identify the packets which comply with the traffic conformance criteria and those which do not, called the "in" and "out" of profile packets, respectively.
- *Traffic Conformance Parameters*, a set of parameters required as input by the *Traffic Conformance Algorithm;* generally speaking, these parameters express the traffic conformance criteria in terms of rate (bandwidth) thresholds.

Basically, this clause includes the information required for configuring the traffic conditioners at the provider edges or border gateways for controlling the traffic injected in the provider domain.

Examples of *Traffic Conformance Algorithms* are: leaky bucket, token bucket, combined token bucket with peak, a two-rate three-colour marker scheme and an MTU-based scheme. Associated *Traffic Conformance Parameters* include: peak rate, token bucket rate, bucket depth and maximum transfer unit (MTU).

4.2.3.1.5 Excess Treatment

The *Excess Treatment* clause describes how excess traffic, i.e. out-of-profile traffic will be processed. The process takes place after the application of the *Traffic Conformance Algorithm* (cf. *Traffic Conformance* clause). It includes the following attributes:

- *Action*, specifying the action to be taken for the excess traffic; it can take the following values: <drop (default) | shape | remark>.
- *Action Parameters*, a set of parameters that may be required by the action taken e.g. for remarking, the DSCP value must be specified and for shaping, the buffer size of the shaper.

4.2.3.1.6 Performance Guarantees

The *Performance Guarantees* clause describes the guarantees on packet transfer performance parameters (metrics) that the provider (agrees to) offers to the packets entitled to the SLS (cf. *Flow Id* clause) within the limits of the SLS geographical/topological span (cf. *Scope* clause). The guarantees to be given are subject to the SLS traffic conformance criteria (cf. *Traffic Conformance* clause); guarantees are given for each of the conformance levels, in case of a multi-level *Traffic Conformance Algorithm*, whereas for out-of-profile no particular guarantees can be given. This clause includes the following self-evident attributes, corresponding to the packet transfer performance metrics against which performance guarantees are given.

- *Delay Guarantees*, specifying the guarantees for the one-way packet delay as measured between specific ingress and egress points crossed by the entitled SLS traffic.
- *Jitter Guarantees*, similar to the above
- *Loss Guarantees*, specifying the guarantees for the packet loss probability; this is defined as the ratio of the lost in-profile packets between specific ingress and egress points and the injected in-profile packets at ingress.
- *Throughput Guarantees*, specifying the guarantees for rate of the traffic delivered, that is, as measured at a specific egress point, counting all packets entitled to the SLS. Note that all packets, independently of their conformance level (in/out-of-profile) contribute to measuring the delivered throughput. Indeed, if a customer (only) wants throughput guarantee for its traffic, then he/she does not care whether in- or out-profile packets are dropped, but is only interested in the overall throughput of its generated packet stream.

It may not be necessary for all above attributes to be specified.

The following aspects underlying the semantics of the above attributes are worth noting:

Performance guarantees can only have meaning within a certain topological scope (cf. *Scope* clause), which is usually designated by couples of ingress and egress points; this scope should be well-defined and understood by both the provider offering the SLS and the customer –end-customer of peering provider.

Delay, jitter and packet loss guarantees refer to the in-profile traffic, conforming traffic injected in the domain, whereas throughput guarantees refer to the overall traffic hit the provider boundary.

The following types of performance guarantees are distinguished: *quantitative* and q*ualitative*. The guarantees to a particular performance parameter are said to be quantitative, if they can be expressed in quantitative, numerical, values. Otherwise, they are said to be qualitative; possible qualitative values, as appropriate as per performance parameter, may include: high, medium, low or red, yellow, green. The quantification of the relative difference between the qualitative values is a matter of provider's policy e.g. 'high' could be twice good as 'medium', which in turn is twice as good as 'low'.

Quantitative performance guarantees are expressed as maximum (worst-case) bounds or as (sets of) percentiles or inverse percentiles, indicating also the granularity period of the associated measurements. The meaning of the values of qualitative performance guarantees and/or their relative difference should be clear to the customers, while it should be backed-up with relevant historical performance data.

Similarly, we can distinguish between *quantitative* and q*ualitative* SLSs. A SLS is said to offer quantitative guarantees, if all the specified guarantees are quantitative; otherwise, is said to offer qualitative guarantees and in the case where no guarantees are specified, the SLS is said to be a *best-effort* SLS.

Finally, the following relationships and dependencies must hold between the information defined in this clause and the information under the *Traffic Conformance* and *Excess Treatment* clauses:

Quantitative delay/jitter/loss guarantees cannot be given unless a rate-based *Traffic Conformance Algorithm* is specified, that is, such guarantees can only be given for in-profile traffic and as such, explicit bandwidth constraints on the offered traffic must have been defined.

For in-profile traffic, loss and throughput guarantees are equivalent and only one of them should be specified. The same holds for qualitative guarantees.

Related to the above, quantitative throughput guarantees, in addition to quantitative loss guarantees, can only be given if excess traffic is remarked, not dropped or shaped.

If quantitative throughput guarantees are only given, then the *Traffic Conformance Algorithm* may not be specified. However, the provider may still wish to protect its domain by requesting for the specification of a *Traffic Conformance Algorithm* e.g. setting a bucket token mechanism to operate such that the average rate of the traffic injected in the domain to almost equal to the guaranteed throughput rate.

4.2.3.2 SSS-T Specifications

SSS-T is specified in terms of the following information elements –clauses-, which are described in the following:

- Subscriber Info
- Subscription Id
- Set of SLS
- Invocation Means
- User Info
- Grade of Service
- Service Activation Info
- Service Schedule
- Availability Guarantees
- Reliability Guarantees

4.2.3.2.1 Subscriber Info

The *Subscriber Info* clause includes the required information to uniquely identify a customer, an end-customer or a peering provider who wishes to use (is requesting) a QoS-based service from the provider. Once the service agreement is in place, the customer becomes a subscriber to the provider.

4.2.3.2.2 Subscription Id

A key, uniquely identifying the agreements established by the provider. It is set by the provider.

4.2.3.2.3 Set of SLS

The set of SLSs, that is, the connectivity legs involved in the QoS-based service. Each SLS should be correct and valid instances of the SSS-T, as specified in section 4.2.3.1.

4.2.3.2.4 Invocation Means

The *Invocation Means* clause describes the procedures and related information for invoking the service.

A service can be invoked either *implicitly*, directly as a result of the establishment of the respective agreement, or *explicitly* based on a well-defined signalling protocol e.g. RSVP, SIP or the PCS-to-PS protocol specified by MESCAL for requesting the establishment of inter-domain QoS MPLS-based tunnels (LSPs).

We further distinguish two types of explicitly invoked services, which need to be supported by suitable signalling protocols: *on-demand* and *partially*. On-demand service invocation denotes a request for using the service as a whole, whereas partial service invocation denotes a request for using certain resources/characteristics associated with the service e.g. bandwidth, number of QoS tunnels; obviously, within the constraints of the overall resources/characteristics agreed in the SLS comprising the service.

Partial invocation is particularly useful for managed bandwidth services, allowing customers to dynamically request that portion of service bandwidth, which they happen to require. In the context of inter-domain agreements, such an invocation method may also be useful, as it would facilitate a more accurate and effective provisioning of the required inter-domain resources (cf. the dynamic pSLS establishment functionality specified in the MESCAL solution). Furthermore, this type of invocation particular suites to the pSLS for establishing QoS MPLS-based tunnels (LSPs); in this case, the service resources are the LSPs. Partial invocation could be alternatively carried out through a sequence of modifications of established service agreements, which obviously presents a burden both to providers and customers.

4.2.3.2.5 User Info

The *User Info* clause includes the required information for uniquely identifying the users of the subscriber who are entitled to invoke the service e.g. user id, password. Obviously, this clause should be specified only in the case of explicitly invoked services.

4.2.3.2.6 Grade of Service

The *Grade of Service* clause describes the guarantees for getting through service invocations.

The specification of such guarantees depends on the nature of the service, the invocation type (on-demand, partial) and the capabilities/policies of the provider. Generally speaking, guarantees for on-demand invoked services could be described in terms of the following parameters: minimum number of simultaneous sessions and acceptance percentage beyond that minimum number. Guarantees for partially-invoked services could be described in terms of a set of confidence levels for using a specific percentage of the totally agreed service resources – as in the SLSs. Get through guarantees could be given in quantitative or qualitative terms.

Obviously, this clause should be specified only in the case of explicitly invoked services.

4.2.3.2.7 Service Activation Info

The term *service activation* denotes the appropriate configurations and provisions that need to be undertaken in the provider domain for making the service available to the customer so that its users can use the service. Service activation is an internal process.

Certain services require that they are activated by the provider in a particular way, so that the (users of the) customer can use them. Examples of such services include: MPLS VPNs offered to end-customers and services for transiting QoS traffic that require the exchange of qBGP messages as in the MESCAL solution. Therefore, the way (method, not how) that the service will be activated may be an essential aspect of service offering and a subject for agreement.

The *Required Activation Method* clause includes the required information for describing the method according to which the service is to be activated –made available to the customer for use. It includes the following attributes:

- *Required Activation Method*, describing the particular method according to which, the service should be made available to the customer for use. It could be specified in terms of a set of URLs or protocols e.g. BGP, qBGP or possible technologies e.g. MPLS VPN and related parameters.
- *Activation Verification Procedures*, describing the procedures necessary to be undertaken for ensuring that the service has been activated as specified above. Its specification is outside the scope of MESCAL investigation, as it relates to the issue of service assurance.

If this clause is specified, the agreement is considered to be in effect according to the agreed *Service Schedule* only after the successful undertaking of the involved *Activation Verification Procedures*.

4.2.3.2.8 Service Schedule

The *Service Schedule* clause describes the time period during which the service should be made available to the customer, in other words the time constraints for using the service. It includes the following attributes:

- *Start Time*
- *Termination Time*
- *Hours*, specifying a range of hours of the specified *Days* of the *Months* during which, the service should be made available to the customer.
- *Days*, specifying a range of days of the specified *Months* during the specified *Hours* of which, the service should be made available to the customer.
- *Months*, specifying a range of months during the specified *Hours* of the *Days* of which, the service should be made available to the customer.

The specification of the exact semantics of the *Start Time* and *Termination Time* or the need for other similar attributes is for further study. The benefit of such information to service agreement management and inter-domain traffic engineering (e.g. could be beneficial in that agreement flapping could be avoided) needs to be investigated thoroughly, while the implication on service negotiations and activation needs to be assessed.

4.2.3.2.9 Availability Guarantees

The *Availability Guarantees* clause includes a single attribute denoting the probability of the service to be made available to the customer as required according to the agreed terms and conditions –in the SSS. It may be specified quantitatively as a percentage or qualitatively e.g. high, medium, low.

4.2.3.2.10 Reliability Guarantees

The Reliability Guarantees clause describes guarantees for reliably providing the service during its lifetime. Its specification is outside the scope of MESCAL investigation.

4.2.4 MESCAL pSLS models

In this section we present suitable models for the different types of pSLS identified by MESCAL (cf. section 4.2.2) according to the different types of business relationships between providers.

While the SLS-T and SSS-T templates, as specified in the previous sections, present an open, detailed model for describing the technical aspects (from connectivity perspectives) of general QoS-based services, including pSLS, there is need for more condensed, summarised pSLS models, for a number of reasons:

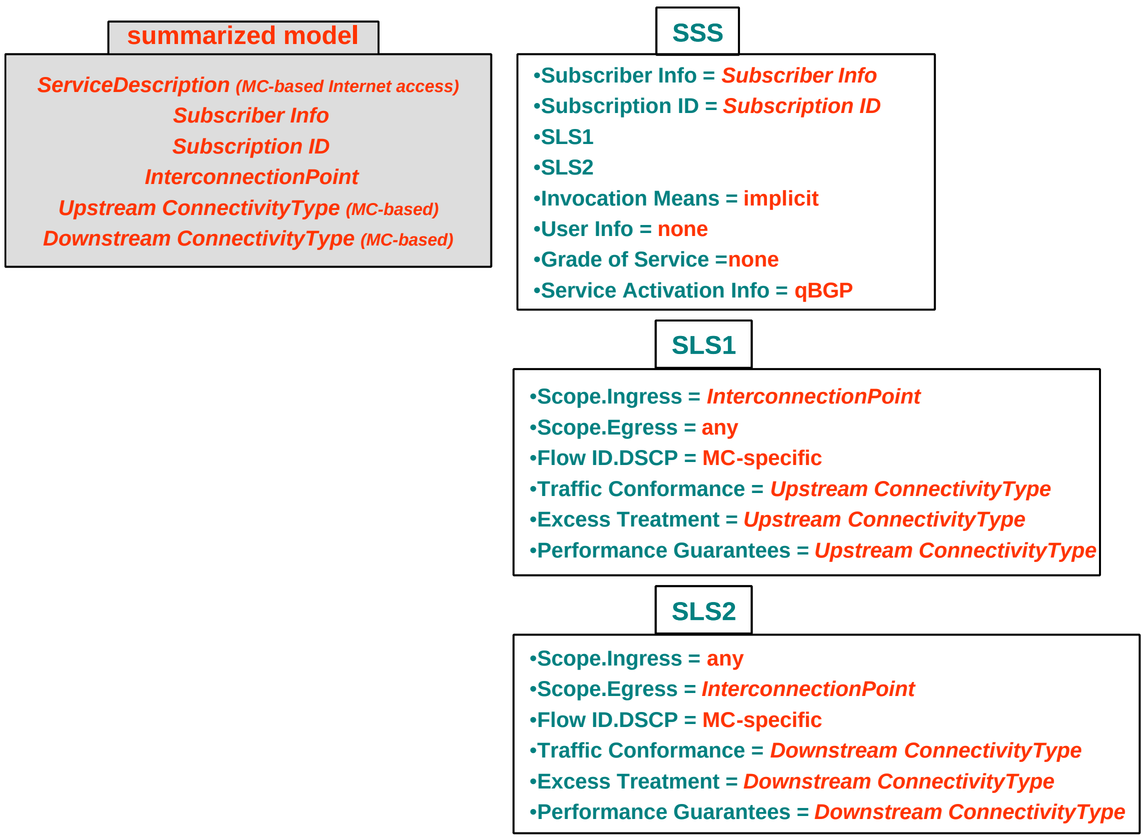
- pSLS need to reflect the specific context of the particular business relationship; not to be expressed in general terms, which might create ambiguity and confusion.
- The service fill-in (subscription) process at the abstraction of SSS-T might be tedious from customer perspectives, because SSS-T specification entails the specification of a number of attributes.
- There are a number of engineering incompatibilities between the values of the attributes of an SSS-T that the customers might not be aware. Therefore, customers might get frustrated as the provider would turn down their service requests, simply because they were not formed correctly.

To the above end, we increase the abstraction level of the SSS-T attributes (as appropriate to the pSLS context), by introducing the so-called *group-alias attributes*. A group-alias attribute is strictly associated with some SSS-TT attributes, providing 'alias' for these SSS-T attributes. By definition, there must be a one-to-one mapping between the values of a group-alias attribute and the values of the SSS-T attributes that is used to alias. Evidently, through the alias mapping function, group-alias attributes can be used to eliminate invalid combinations of SSS-T attribute values. Group-alias attributes may be complex, in the sense of containing other group-alias attributes.

For pSLS, the following self-evident group-alias attributes have been identified:

- *ServiceDescription*, providing alias to *Invocation Means* and *Service Activation Info* attributes,
- *InterconnectionPoint*, providing alias to *Scope* attributes,
- *DestinationNets*, providing alias to *Scope* and *Flow Id* attributes,
- *Site*, providing alias to *Scope* and *Flow Id* attributes,
- *ConnectivityType*, including performance guarantees and bandwidth, providing alias to *Traffic Conformance*, *Excess Treatment* and *Performance Guarantees* attributes

The figures below depict the summarised models of the different types of pSLS identified. Note that the Service Schedule, Availability Guarantees and Reliability Guarantees attributes of the SSS-T are common to both summarised and detailed pSLS models and they are not depicted in the figures.



summarized model

ServiceDescription *(MC-based Internet access)*
Subscriber Info
Subscription ID
InterconnectionPoint
Upstream ConnectivityType *(MC-based)*
Downstream ConnectivityType *(MC-based)*

SSS

•Subscriber Info = *Subscriber Info*
•Subscription ID = *Subscription ID*
•SLS1
•SLS2
•Invocation Means = implicit
•User Info = none
•Grade of Service =none
•Service Activation Info = qBGP

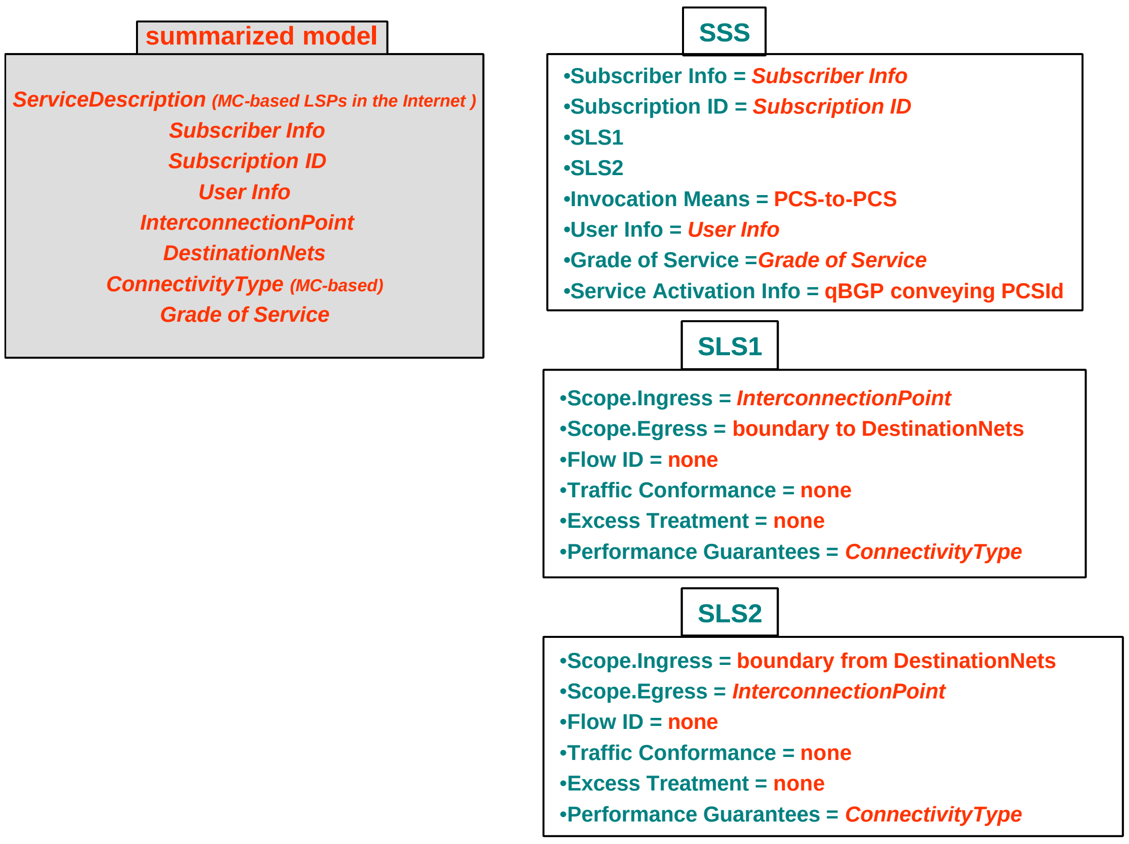
SLS1

•Scope.Ingress = *InterconnectionPoint*
•Scope.Egress = any
•Flow ID.DSCP = MC-specific
•Traffic Conformance = *Upstream ConnectivityType*
•Excess Treatment = *Upstream ConnectivityType*
•Performance Guarantees = *Upstream ConnectivityType*

SLS2

•Scope.Ingress = any
•Scope.Egress = *InterconnectionPoint*
•Flow ID.DSCP = MC-specific
•Traffic Conformance = *Downstream ConnectivityType*
•Excess Treatment = *Downstream ConnectivityType*
•Performance Guarantees = *Downstream ConnectivityType*

Figure 19. Model of pSLS for Internet access at loose QoS



summarized model

ServiceDescription *(MC-based LSPs in the Internet)*
Subscriber Info
Subscription ID
User Info
InterconnectionPoint
DestinationNets
ConnectivityType *(MC-based)*
Grade of Service

SSS

•Subscriber Info = *Subscriber Info*
•Subscription ID = *Subscription ID*
•SLS1
•SLS2
•Invocation Means = PCS-to-PCS
•User Info = *User Info*
•Grade of Service =*Grade of Service*
•Service Activation Info = qBGP conveying PCSId

SLS1

•Scope.Ingress = *InterconnectionPoint*
•Scope.Egress = boundary to DestinationNets
•Flow ID = none
•Traffic Conformance = none
•Excess Treatment = none
•Performance Guarantees = *ConnectivityType*

SLS2

•Scope.Ingress = boundary from DestinationNets
•Scope.Egress = *InterconnectionPoint*
•Flow ID = none
•Traffic Conformance = none
•Excess Treatment = none
•Performance Guarantees = *ConnectivityType*

Figure 20. Model of pSLS for loose QoS tunnels in the Internet

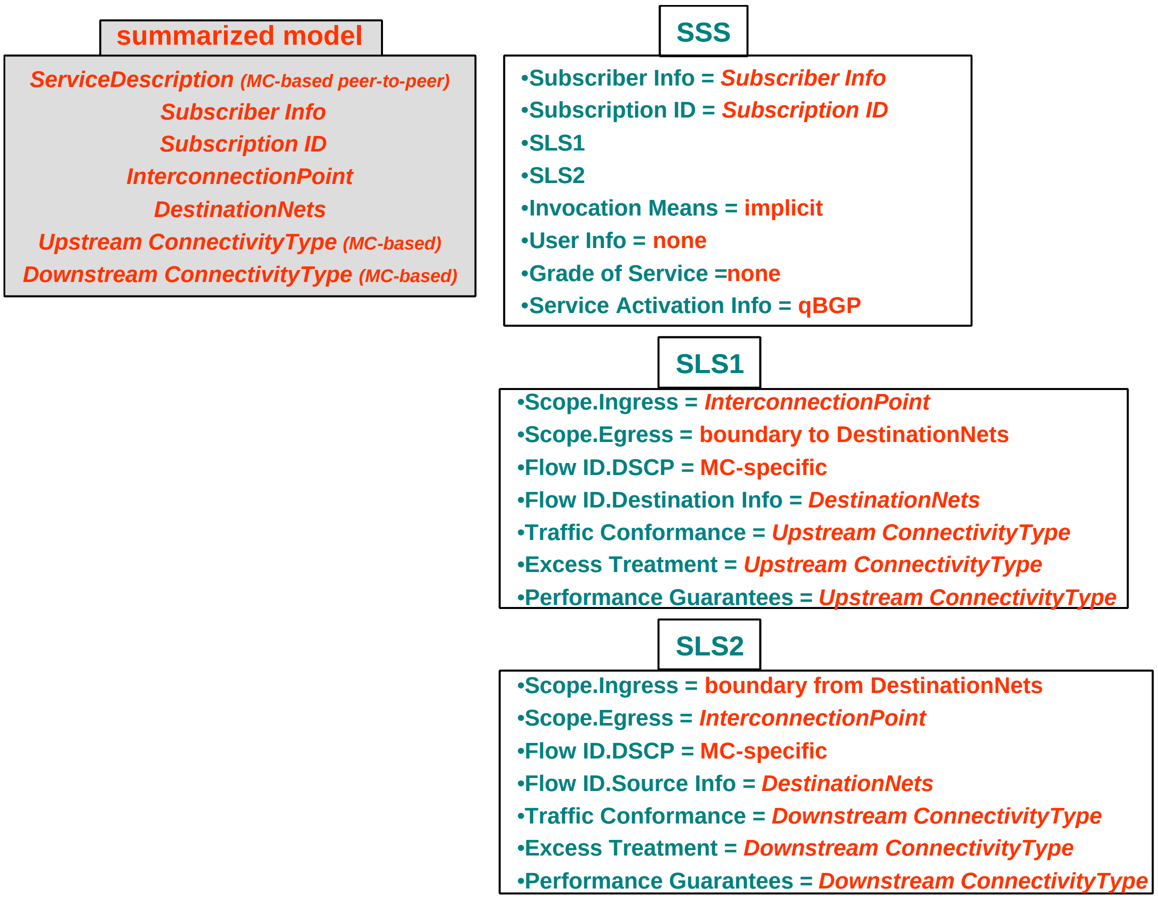

summarized model
ServiceDescription (MC-based peer-to-peer)
Subscriber Info
Subscription ID
InterconnectionPoint
DestinationNets
Upstream ConnectivityType (MC-based)
Downstream ConnectivityType (MC-based)
SSS
•Subscriber Info = Subscriber Info
•Subscription ID = Subscription ID
•SLS1
•SLS2
•Invocation Means = implicit
•User Info = none
•Grade of Service =none
•Service Activation Info = qBGP
SLS1
•Scope.Ingress = InterconnectionPoint
•Scope.Egress = boundary to DestinationNets
•Flow ID.DSCP = MC-specific
•Flow ID.Destination Info = DestinationNets
•Traffic Conformance = Upstream ConnectivityType
•Excess Treatment = Upstream ConnectivityType
•Performance Guarantees = Upstream ConnectivityType
SLS2
•Scope.Ingress = boundary from DestinationNets
•Scope.Egress = InterconnectionPoint
•Flow ID.DSCP = MC-specific
•Flow ID.Source Info = DestinationNets
•Traffic Conformance = Downstream ConnectivityType
•Excess Treatment = Downstream ConnectivityType
•Performance Guarantees = Downstream ConnectivityType

Figure 21. Model of pSLS for traffic inter-exchange at a loose QoS

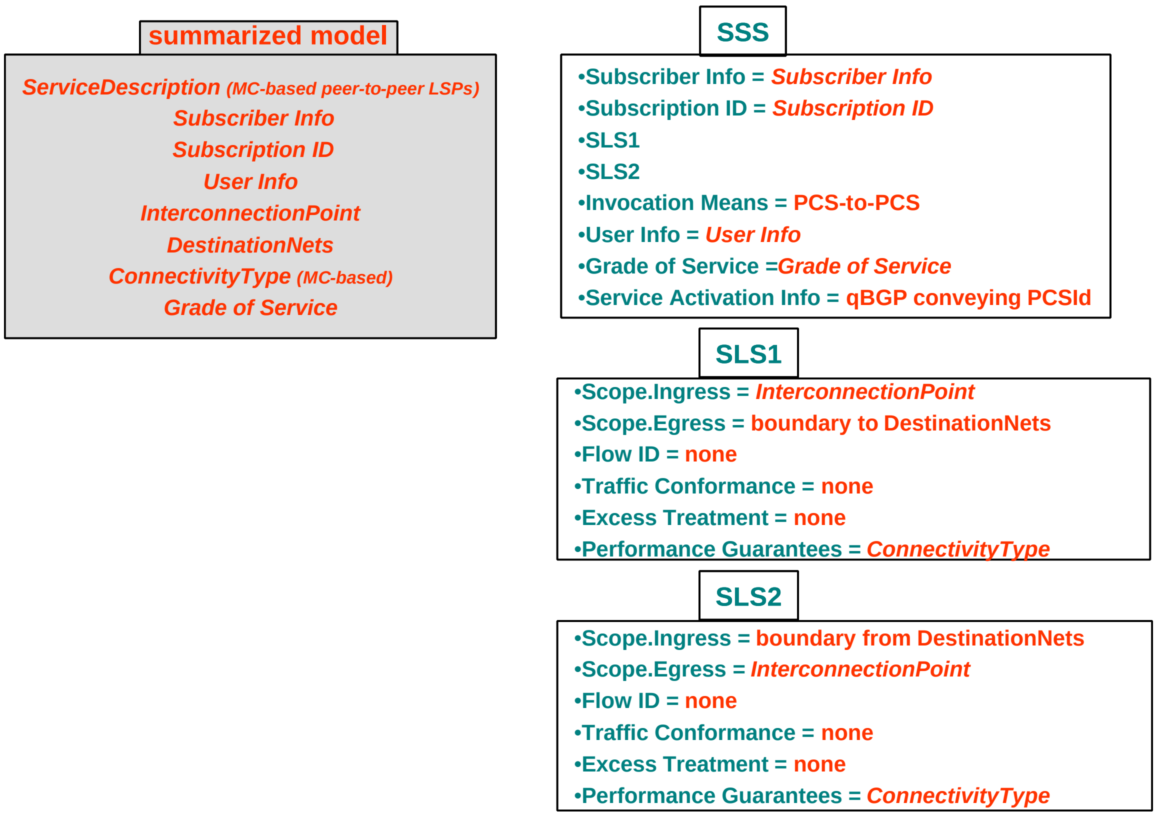

summarized model
ServiceDescription (MC-based peer-to-peer LSPs)
Subscriber Info
Subscription ID
User Info
InterconnectionPoint
DestinationNets
ConnectivityType (MC-based)
Grade of Service
SSS
•Subscriber Info = Subscriber Info
•Subscription ID = Subscription ID
•SLS1
•SLS2
•Invocation Means = PCS-to-PCS
•User Info = User Info
•Grade of Service =Grade of Service
•Service Activation Info = qBGP conveying PCSId
SLS1
•Scope.Ingress = InterconnectionPoint
•Scope.Egress = boundary to DestinationNets
•Flow ID = none
•Traffic Conformance = none
•Excess Treatment = none
•Performance Guarantees = ConnectivityType
SLS2
•Scope.Ingress = boundary from DestinationNets
•Scope.Egress = InterconnectionPoint
•Flow ID = none
•Traffic Conformance = none
•Excess Treatment = none
•Performance Guarantees = ConnectivityType

Figure 22. Model of pSLS for loose QoS tunnel extension

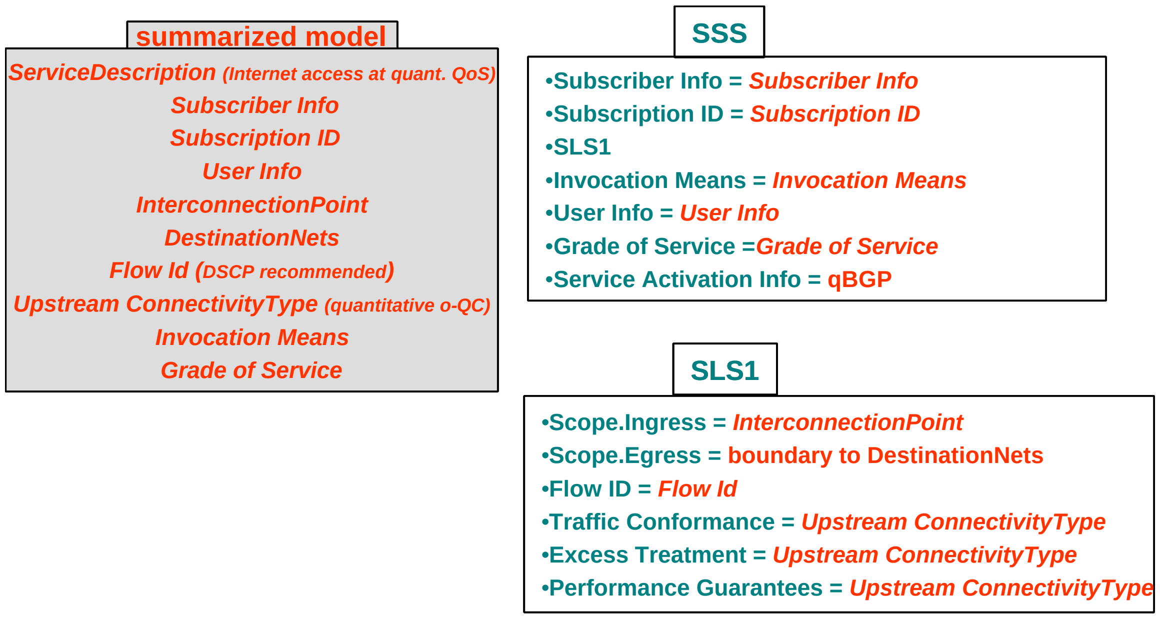

summarized model
ServiceDescription (Internet access at quant. QoS)
Subscriber Info
Subscription ID
User Info
InterconnectionPoint
DestinationNets
Flow Id (DSCP recommended)
Upstream ConnectivityType (quantitative o-QC)
Invocation Means
Grade of Service
SSS
•Subscriber Info = Subscriber Info
•Subscription ID = Subscription ID
•SLS1
•Invocation Means = Invocation Means
•User Info = User Info
•Grade of Service =Grade of Service
•Service Activation Info = qBGP
SLS1
•Scope.Ingress = InterconnectionPoint
•Scope.Egress = boundary to DestinationNets
•Flow ID = Flow Id
•Traffic Conformance = Upstream ConnectivityType
•Excess Treatment = Upstream ConnectivityType
•Performance Guarantees = Upstream ConnectivityType

Figure 23. Model of pSLS for Internet access at a statistically guaranteed QoS

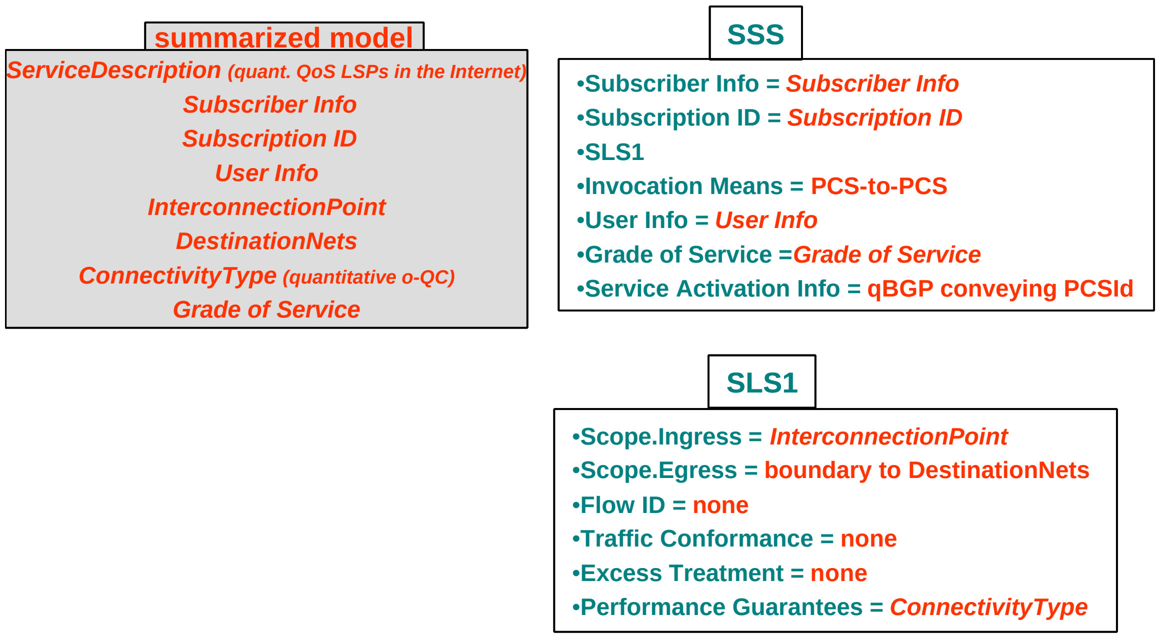

summarized model
ServiceDescription (quant. QoS LSPs in the Internet)
Subscriber Info
Subscription ID
User Info
InterconnectionPoint
DestinationNets
ConnectivityType (quantitative o-QC)
Grade of Service
SSS
•Subscriber Info = Subscriber Info
•Subscription ID = Subscription ID
•SLS1
•Invocation Means = PCS-to-PCS
•User Info = User Info
•Grade of Service =Grade of Service
•Service Activation Info = qBGP conveying PCSId
SLS1
•Scope.Ingress = InterconnectionPoint
•Scope.Egress = boundary to DestinationNets
•Flow ID = none
•Traffic Conformance = none
•Excess Treatment = none
•Performance Guarantees = ConnectivityType

Figure 24. Model of pSLS for statistically guaranteed QoS tunnels in the Internet

4.3 Service Negotiation protocol

4.3.1 Introduction

Quality of service (QoS) delivery across the Internet provides new business opportunities, but also presents new challenges. Nowadays, QoS-based services are offered on the basis of the so-called *Service Level Agreements (SLAs)*, which set the terms and conditions on behalf of both providers and customers in providing and requesting services, respectively.

It is without doubt that flexibility and automation are the 'names of the game' in QoS-based service provisioning. Providers have to be able to offer quickly new services according to market needs, while ensuring that their networks are appropriately configured to efficiently deliver the quality requirements of the services offered. At the same time, customers should be able to find out the offered QoS-based services and subsequently request/modify the level of QoS they desire according to their actual needs.

As flexibility requirements are high, SLAs are not necessarily monolithic contracts -agreed once, valid forever. Although this might hold for a number of business cases e.g. between peer wholesalers, SLAs might well be short-lived agreements e.g. SLAs between customers and a provider for pay-per-view services over the weekend. Furthermore, SLAs should be seen as 'living documents', in the sense of being modified, on a common agreement basis, according to actual customer needs and providers' availability or related policies. In line with this view of SLAs, is the widely accepted distinction between static and dynamic SLAs [BLAK98], [NIC99].

In the above scenery then, automated means for agreeing on SLAs (establishment, modification, deletion) will pave the way towards flexible and automated QoS-based service provisioning. At the same time it will constitute an important step in the evolution of the Internet itself. Compared to agreeing on SLAs in a sort of a manual fashion (e.g. through fax, e-mail, post), as it is mainly the practice of today, automated establishment of SLAs has a number of benefits to both providers and customers. For providers, it reduces operational costs, contributes to an integrated, fully automated service provisioning process and increases the level of attraction of the offered services. For customers, it increases their flexibility in requesting and accessing services by reducing the required time. Furthermore, automated means for SLA agreement opens-up new ways, and businesses, in promoting QoS-based services in the Internet. E.g. through Web-based service portals, where customers can view existing service offerings and agree on SLAs for desired services according to actual needs.

To automate the process of agreeing on SLAs new protocols are required. These protocols should enable customers and providers or peer providers to automatically negotiate between each other with the purpose to finally agree on a SLA. We call these protocols SLA negotiation protocols.

We view that there is a clear distinction between SLA negotiation protocols and QoS-signalling or reservation or QoS-enabled session control protocols (e.g. RSVP, SIBBS, SIP, H.323, PPP). Specifically, we view that SLA negotiation protocols are used for agreeing on SLAs, whereas, QoS-signalling or reservation or session control protocols are used for signalling/requesting the level of QoS that customers require, should respective SLAs with the providers have been agreed. SLA negotiation protocols operate at service subscription epochs, where customers subscribe to the desired services offered by the providers, and QoS-signalling or reservation or session control protocols operate at service invocation epochs, where the users of the customers (subscribers) call for the services to which have been subscribed. The distinction between service subscription and invocation is required mainly for AAA (authentication, authorisation and accounting) purposes i.e. for checking conformance of user service requests against agreed profiles, which is essential in SLA-based service provisioning. This view largely follows current business practices; it is also in line with the principles of the service management framework presented in [GOD02a].

In the above spirit, a protocol, the Service Negotiation Protocol (SrNP), for SLA negotiations is presented. It should be noted that SrNP is not specific to the particular contents of SLAs, nor it is specific to particular transport, policy or information exchange protocols. Furthermore, SrNP is completely decoupled from the negotiation logic -the logic, per negotiating party, for conducting negotiations- offering to it, through clear interfaces, the necessary primitives required for enabling negotiations. These features increase protocol applicability, and as such, SrNP could be used for establishing any type of agreements (e.g. on price-lists) in a general e-commerce context.

4.3.2 Negotiation Protocol Requirements

SrNP design is driven by the following requirements. It should be noted that these requirements are drawn from our own experience and objectives, as requirements for negotiation protocols have not yet been commonly agreed.

Functional requirements

- The negotiation protocol should provide for *primitives to enable the process of negotiations* between two or more parties (negotiating parties). Generally speaking, the negotiation process is a process where several parties are seeking for an agreement on a number of commonly understood issues e.g. on a SLA.
- There should be a clear distinction between the primitives offered by the *negotiation protocol* and the *negotiation logic*. The term 'negotiation logic' denotes the logic according to which negotiations are conducted on behalf of each negotiating party. Negotiation logic should be specific to each negotiation party, being subject to its business policies and operational capabilities, and as such, is considered application- and domain-specific. In essence, the negotiation protocol should provide a service to the negotiation logic i.e. it should be seen as a layer based on which application-specific negotiation logic could be built.
- The negotiation protocol should not duplicate but complement the functionality of existing, widely deployed, standardised protocols. For SLA agreements (QoS negotiation), the corresponding negotiation protocols should not duplicate (aspects of) the functionality of existing QoS-signalling or reservation or session control protocols (e.g. RSVP).
- The negotiation protocol should lead at convergent negotiation processes. Appropriate mechanisms should be provided at protocol layer, for ensuring that the negotiation process can terminate successfully or unsuccessfully in finite steps and in a reasonable time period, as deemed necessary by (the negotiation logic of) each of the negotiating parties.
- The negotiation protocol should be independent of the underlying transport and network protocols. In fact, it should be able to operate with multiple such protocols.

Non-functional requirements

- The negotiation protocol should provide for secure and reliable communication.
- The negotiation protocol should be expandable in terms of additional negotiation primitives.
- The negotiation protocol should be able to support a number of simultaneous active negotiation processes.

It is clear, that the above requirements contribute to the openness and therefore the applicability of negotiation protocols; they are not specific to SLA negotiations and they could apply to any negotiation protocol.

4.3.3 Negotiation Model

The following assumptions underline the negotiation model to which SrNP has been designed to apply.

It is assumed that the negotiation process involves *two parties* only; one acting in a server role, called the *server*, and the other acting in a client role, called the *client*. The roles are exclusive to the parties that is, a party cannot act in both roles in the context of a particular negotiation process. Following the usual distinction between client and server roles (client requests, server responds), in the context of a negotiation process, these roles are distinguished in that agreements can only be pursued by the client towards the server. This distinction is in line with the semantics underlying a customer-provider relationship between two interacting parties. Valid client-server tuples of negotiating parties could be: customer-provider (intra-domain SLA negotiations) or provider-provider (inter-domain SLA negotiations). Note that the provider-customer tuple is also considered valid. For instance, this case may arise in situations where the provider deems necessary to renegotiate SLAs with some customers for improving or lowering the quality of the subscribed services.

It is assumed that the issues under negotiation can be described in a form of a *document*. The target of the negotiation process is then for the negotiating parties to come to an *agreement regarding the content of* (information included in) the document.

Furthermore, it is assumed that all negotiating parties have a common understanding of the semantics and syntax of the information included in the document as well as means for constructing, extracting and manipulating the information in the document. In line with the requirements presented in the previous section, document/information format, construction and manipulation are not of concern to the protocol, but rather of the negotiation logic. Evidently then, SrNP is not specific to any SLA format or to the content of SLAs. It is general enough to apply to negotiating any issues, provided that these issues can be appropriately described in the form of a commonly understood document.

Finally, it is assumed that authentication and authorisation with respect to negotiation aspects are not of concern to SrNP; they are of concern to the negotiation logic that SrNP services. It is also assumed that SrNP uses the services of a reliable and secure transport protocol.

4.3.4 SrNP Overview

SrNP is an *application-layer*, *session-oriented* protocol allowing for sessions to:

- establish an agreement,
- modify an established agreement, and to
- delete an established agreement

SrNP sessions are initiated by the client.

Agreement Establishment Session

Generally speaking, the negotiation process for establishing an agreement is an iterative process, whereby the negotiating parties exchange their views/requirements on the issues under negotiation until an agreement is reached.

SrNP follows a *client-server*, *dialogue-based* approach for realising the necessary interactions between the negotiating parties towards establishing an agreement. Specifically:

First, the client *connects* to the server to initiate a session for negotiating the establishment of an agreement. Subsequently, SrNP allows for the client to issue *proposals* and the server to respond by issuing *revisions* or by calling for an *agreement*. Proposals and revisions convey the client's and the server's views/requirements on the issues under negotiation, respectively. Through revisions, the server is enabled to respond to the client views/requirements not in a monolithic 'agree/do not agree' manner but, in a flexible, in the spirit of 'I could agree provided that/even if', manner indicating the points of argumentation and suggesting possible alternatives. It is up to the negotiation logic of the client to determine whether to adhere or not to adhere to the suggested revisions in subsequent proposals. The client and the server exchange proposals and revisions respectively until the server responds with (calls for) an *agreement* on the last sent proposal and the client *accepts* it, or either party *rejects* the negotiation process. At these points the negotiation process concludes successfully or unsuccessfully, respectively. When calling for an agreement the server also includes the last received proposal by the client as a form of 'hand-shaking'.

The client and the server interact in a dialogue (half-duplex) manner; once a party sends information to the other party, the party is blocked until a *valid response* from the other party is received. Specifically, once the client sends a proposal, it is blocked until it receives a revision or an agreement or a rejection from the server. Similarly, once the server sends a revision or calls for an agreement, it is blocked until it receives an alternative proposal or an acceptance on the called agreement or a rejection from the client. Once a party issues a rejection, the protocol terminates the negotiation process from this party, without waiting any further response from the other party.

To ensure graceful operation, SrNP does not allow a party to be blocked forever, waiting to receive a valid response from the other party. To this end, when a party sends information to the other party, SrNP requires that the party must specify a maximum tolerable time period willing to wait for the other party to respond back. SrNP will reject the negotiation process on behalf of the sending party, if during the specified maximum tolerable time period no valid response from the other party is received. In addition to avoiding communication blocking, this mechanism of SrNP has the intuitive counterpart of 'sent information is only valid for a specific time period'.

SrNP also offers the negotiation features of 'take it or leave it' (or, 'last word') and 'please wait to be served'. Specifically:

SrNP allows for distinguishing proposals and revisions as *ordinary* and *last*. Opposed to ordinary, last proposals and last revisions cannot be followed by other ordinary revisions and proposals respectively, but only by 'commitments' (definite responses). Specifically, a last proposal forces the server to respond with an agreement or a rejection, and a last revision forces the client to respond with a last proposal or a rejection. In addition to its intuitive counter part ('take it or leave it' or 'last word'), this feature offers a lever for enforcing the termination of a negotiation process in finite steps.

SrNP allows for the server to request that delays its response to a client's proposal beyond the maximum tolerable time period the client is willing to wait for. This can only happen when the server sees that an agreement is likely to be reached shortly after the elapse of the time period specified by the client. As such, if the client confirms, the server can only respond by calling for an agreement or rejecting the negotiation process.

Agreement Modification Session

During this session, SrNP operates similarly to the agreement establishment session outlined above. In this case, the first proposal to be sent by the client denotes the agreement modifications that the client wishes to make.

Agreement Deletion Session

Once the client has successfully initiated a session for deleting an already established agreement, SrNP allows for the server to respond by either accepting or rejecting the agreement deletion request. The decision for accepting or rejecting the deletion request is taken by the server negotiation logic.

4.3.5 SrNP Messages and Interface

4.3.5.1 Protocol Messages

The SrNP messages reflect the negotiation primitives offered by the protocol to the negotiation logic. Two types of SrNP messages are distinguished: *client messages*, sent only by the client, and *server messages*, sent only by the server.

Following the dialogue nature of SrNP, during a negotiation session initiated by the client for establishing/modifying/deleting an agreement, client and server messages are exchanged alternately (one after the other); server messages are sent in response to client messages and vice versa.

Furthermore, SrNP dictates that messages must be exchanged in a particular order, reflecting the natural evolution of a negotiation process. That is, given a message sent by a party, the other party can respond only with specific messages, which SrNP regards as *valid responses* to the message sent. Subsequently, the party, which has sent a message and received a valid response by the other party, can only send specific messages corresponding to the valid responses of the response-message received, and so on until the negotiations are terminated.

The SrNP messages are described in Table 2 and their parameters in Table 3.

SrNP Client Messages		
Message	**Description**	**Valid Responses (from the server)**
`SessionInit`	It requests the initiation of a session for negotiating the establishment/modification/deletion of an agreement. It is the first message that the client must send.	`Accept` `Reject`
`Proposal`	It carries the client's requirements/views on the issues under negotiation, described in a form of document, which must be commonly understood by the negotiating parties. This message is exchanged during negotiation sessions initiated for establishing or modifying an agreement. The carried document is constructed by the client's negotiation logic and the semantics and syntax of the information included in it are transparent to the protocol. This -or the `LastProposal` message- is the first message that the client must send after the negotiation session has been established.	`Revision` `LastRevision` `ProposalOnHold` `AgreedProposal` `Reject`
`LastProposal`	A `Proposal` that forces the server to accept or not accept the client's requirements/views on the issues under negotiation carried by the message.	`ProposalOnHold` `AgreedProposal` `Reject`
`AcceptToHold`	It confirms the server's request (cf. `ProposalOnHold` message) to delay its response to the last sent client's `Proposal/LastProposal` message.	`AgreedProposal` `Reject`
`Accept`	It indicates that the client accepts the agreement called by the server (cf. `AgreedProposal` message). When reliably delivered to the server, the protocol terminates at both ends concluding successfully the negotiations.	None.
`Reject`	It indicates that the client cannot accept the last received server's response and, as such, is not willing to continue the negotiations. When reliably delivered to the server, the protocol terminates at both ends concluding unsuccessfully the negotiations.	None.
SrNP Server Messages		
Message	**Description**	**Valid Responses (from the client)**
`Revision`	It carries the server's counter-requirements/views on the issues under negotiation, should the server cannot accept (some of) the respective client's requirements/views as last received (cf. `Proposal` message). Server's counter- requirements/views are described in a form of a document, constructed by the server's negotiation logic, which must be commonly understood by the negotiating parties. The semantics and syntax of the information included in the document are transparent to the protocol.	`Proposal` `LastProposal` `Reject`
`LastRevision`	A `Revision` that forces the client to respond in a definite manner; to adhere to the server's counter-requirements/views or insist on its own views/requirements or abort the negotiations (cf. `Proposal/LastProposal, Reject` messages).	`Proposal` `LastProposal` `Reject`
`AgreedProposal`	It indicates that the server accepts the last received client's requirements/views on the issues under negotiations (cf. `Proposal/LastProposal` messages), therefore calling for an agreement. The message should carry the last received document by the client as a form of 'hand-shaking'.	`Accept` `Reject`
`ProposalOnHold`	It requests that the server can respond to the last received client's requirements/ views on the issues under negotiations (cf. `Proposal/LastProposal` messages) within a specific time period in the near future. However, it implies that the response to be given at that time should be definite; either call for an agreement (cf. `AgreedProposal` message) or not (cf. `Reject` message).	`AcceptToHold` `Reject`
`Accept`	It confirms the client's request to initiate a negotiation session for agreement establishment/modification/deletion (cf. `SessionInit` message).	`Proposal` `LastProposal` `Reject`
`Reject`	It indicates that the server cannot accept the last received client's request/response and, as such, is not willing to continue the negotiations. When reliably delivered to the client, the protocol terminates at both ends concluding unsuccessfully the negotiations.	None.

Table 2. SrNP protocol messages.

SrNP Header Parameters –common to all messages		
Parameter	**Description**	
messageId	Unique identification of the sent messages in the locality of a party. It is set by the SrNP mechanism.	
inResponseTo	The messageId of the message to which this message is sent as a response. By examining this parameter a party is able to determine whether a received valid response is *correct* i.e. it indeed corresponds to the last message sent by this party. It is set by the SrNP mechanism.	
timeToRespond	The maximum time period, in minutes, that a party sending a message is willing to wait for the other party to respond. If this period elapses, the protocol terminates from the sending party by issuing a Reject message. It is specified by the negotiation logic.	
SrNP Client Message Parameters		
Message	**Parameter**	**Description**
SessionInit	SessionType	The type of the negotiation session the client wishes to initiate. It takes three possible values: newAgreement, modifyAgreement, deleteAgreement.
	AgreementId	In cases of sessions for modification/deletion of established agreements, it is the unique identifier of the agreement to be modified or deleted as set by the server (cf. AgreedProposal message). In cases of sessions for establishing a new agreement, it is left unspecified.
Proposal	ProposedDocument	The document describing the client's requirements/views on the issues under negotiations. It is constructed by the client's negotiation logic.
	selectedAlternativeId	The identification of the alternative (see below) suggested by the server in the last received Revision message, to which the proposedDocument may adhere. This parameter could be left unspecified. In any case, it is specified by the client's negotiation logic.
LastProposal	As in Proposal.	
AcceptToHold	None.	
Accept	None.	
Reject	Reason	The reason for aborting the negotiation process. It is specified by the client's negotiation logic.
SrNP Server Messages		
Message	**Parameter**	**Description**
Revision	alternatives	It describes the server's counter-requirements/views on the issues under negotiation based on the respective client's requirements/views, as last received by a Proposal/LastProposal message. The server counter-requirements/views may be expressed as a set of exclusive alternatives. As such, this parameter, specified by the server's negotiation logic, is a compound one including a list of the following parameters:
	alternativeId	The unique identification of a specific alternative (see below) within a Revision message.
	validUntil	The date and time that the specific alternative (see below) expires.
	alternative	A document describing a specific server alternative that is, a particular set of server's counter-requirements/views on the issues under negotiation.
LastRevision	As in Revision.	
AgreedProposal	agreementId	A unique identifier of an agreement called by the server. It is specified by the server's negotiation logic.
	agreedDocument	The last received document describing the client's requirements/views on the issues under negotiation (carried by a Proposal/LastProposal message), based on which the server called for an agreement.
ProposalOnHold	timeToHoldOn	The maximum time period, in minutes, within which the server anticipates to issue its response (which, will be definite) to the last received client's requirements/views on the issues under negotiation (carried by a Proposal/LastProposal message). Apparently, this should be greater than the time period the client can possibly wait for receiving server's response (cf. timeToRespond parameter of the message header). It is specified by the server's negotiation logic.
Accept	None.	
Reject	Reason	The reason for aborting the negotiation process. It is specified by the server's negotiation logic.

Table 3. Parameters of the SrNP protocol messages.

4.3.5.2 Message Sequence Charts (MSCs)

Figure 25 and Figure 26 depict the exchange of SrNP protocol messages in a number of typical negotiation processes initiated for agreement establishment or modification.

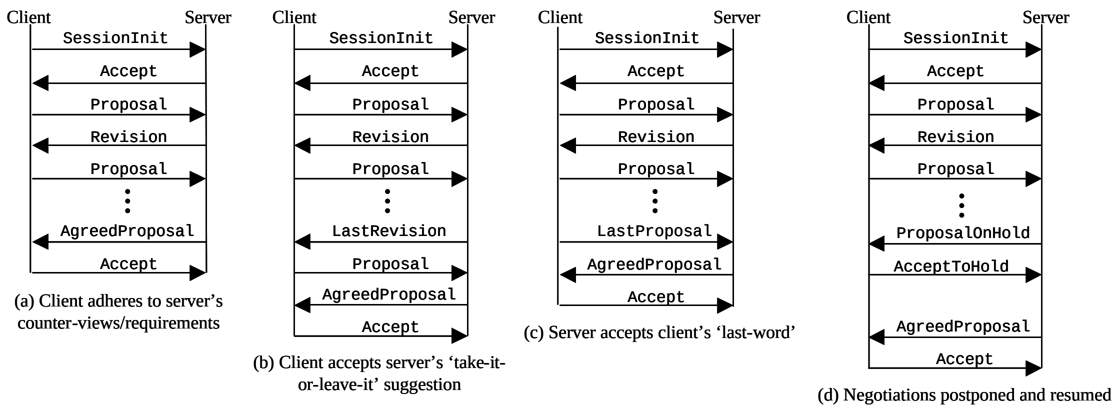


Figure 25. MSCs of successful negotiations

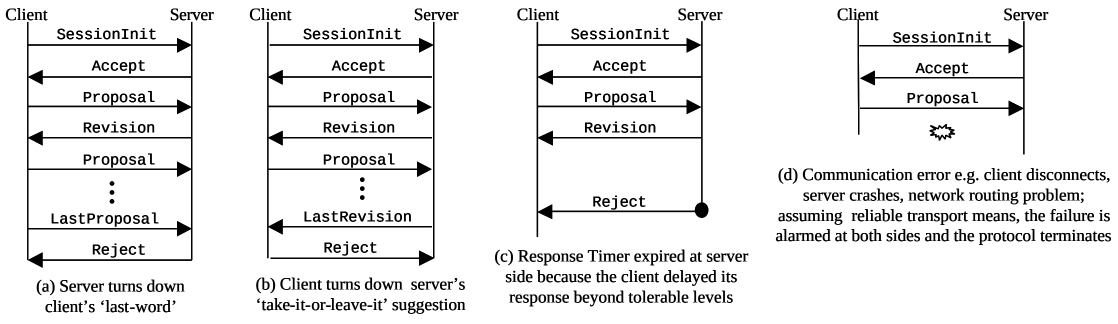

Client
Server
SessionInit
Accept
Proposal
Revision
Proposal
LastProposal
Reject
(a) Server turns down client's 'last-word'
Client
Server
SessionInit
Accept
Proposal
Revision
Proposal
LastRevision
Reject
(b) Client turns down server's 'take-it-or-leave-it' suggestion
Client
Server
SessionInit
Accept
Proposal
Revision
Reject
(c) Response Timer expired at server side because the client delayed its response beyond tolerable levels
Client
Server
SessionInit
Accept
Proposal
(d) Communication error e.g. client disconnects, server crashes, network routing problem; assuming reliable transport means, the failure is alarmed at both sides and the protocol terminates

Figure 26. MSCs of unsuccessful negotiations

4.3.5.3 Interface (I/F) Messages

The interface that SrNP offers to applications realising negotiation logic is described in a technology-independent manner through a set of messages depicted in Table 4.

SrNP interface (I/F) messages are self-explained and mainly correspond to the SrNP protocol messages presented previously. The prefix "Send" denotes the 'pull'-part of the SrNP interface allowing the application to send a protocol message to the other party, whereas the prefix "Forward" denotes the 'push'-part of the SrNP interface notifying the application of a protocol message received from the other party.

The `ForwardException` message notifies the application on abnormal protocol termination. This can occur (a) when the maximum tolerable time period that a party has specified to wait for the other party to respond elapses without receiving such a response, (b) on transport service failures and (c) on unexpected application behaviour. The latter occurs when the negotiation logic of a party responds to the last message received from the other party with a not valid response message or with multiple valid response messages.

SrNP Client IF Message	SrNP Server IF Message
`SendConnectToServer`	
`SendSessionInit`	`SendRevision`
`SendProposal`	`SendLastRevision`
`SendLastProposal`	`SendAgreedProposal`
`SendAcceptToHold`	`SendProposalOnHold`
`SendAccept`	`SendAccept`
`SendReject`	`SendReject`
`ForwardRevision`	`ForwardSessionInit`
`ForwardLastRevision`	`ForwardProposal`
`ForwardAgreedProposal`	`ForwardLastProposal`
`ForwardProposalOnHold`	`ForwardAcceptToHold`
`ForwardAccept`	`ForwardAccept`
`ForwardReject`	`ForwardReject`
`ForwardProtocolException`	`ForwardProtocolException`

Table 4. SrNP interface messages

4.3.6 SrNP Finite State Machine (FSM)

We distinguish two types of SrNP FSMs: *client and server FSMs*. In the context of a particular negotiation process, there is a single instance of a client and server FSM at client and server sides respectively. As such, at a client side there may be as many client FSMs as the negotiation processes initiated by the client, and at a server side there may be as many server FSMs as the negotiation processes of the connected clients.

4.3.6.1 Timers

- `ResponseTimer`: SrNP starts this timer whenever a negotiating party sends a message to the other party. Its value, time to expire, is set in accordance to the maximum time period the party can possibly wait for the other party to respond, which should be determined by the party's negotiation logic. SrNP also assigns timer's value to the `timeToRespond` field (see Table 3) of the header of the message to be sent. The timer expires should its time to expire elapses without having received any *valid* (cf. rightmost column of Table 2) and *correct* (cf. `messageId` header parameter, Table 3) *response* message from the other party. At this point, SrNP terminates the negotiation process on behalf of the sending party. Evidently, by utilising this mechanism SrNP avoids communication blocking, while ensures that the negotiation process will terminate in finite steps and time.

4.3.6.2 Events

Considering a particular negotiating party (client or server), in addition to the protocol and interface messages (see Table 2and Table 4), the following events are also considered by the SrNP FSMs:

- `ResponseTimerExpired:` It is fired whenever the `ResponseTimer` expires.
- `TransportError:` It is fired whenever SrNP is notified by the underlying transport services of failures in communicating with the other party.

The above events are considered protocol operation exceptions and are encapsulated into the `ForwardProtocolException` protocol interface message (cf. Table 4). Furthermore, the server FSM considers the following event:

- `ConnectionAccepted:` It is fired whenever the underlying transport services at the server side accept the request of a remote party (client) for establishing a new connection/session at the transport layer (cf. `SendConnectToServer` protocol client interface message, Table 4).

4.3.6.3 SrNP Client FSM

The client FSM includes the following states:

- *Idle:* In this state the protocol is (has become) inactive, waiting for the client's negotiation logic to initiate a new negotiation process.
- *WaitToFormClientMessage (W_FCM):* In this state the protocol waits for the client's negotiation logic to determine and formulate its response to the last message received from the server. While at this state, any other message that may come from the server is 'blocked' in the sense that is not forwarded to the client's negotiation logic.
- *WaitServer:* In this state the protocol waits to receive from the server a response to the last sent message determined by the client's negotiation logic. While at this state, the protocol 'blocks' client's negotiation logic in the sense that is not allowed to send to the server any other message that may come from the client's negotiation logic.

The state transition diagram of the client FSM is shown in Figure 27. The associated actions are depicted in Table 5 and described in pseudo-code in Table 6.

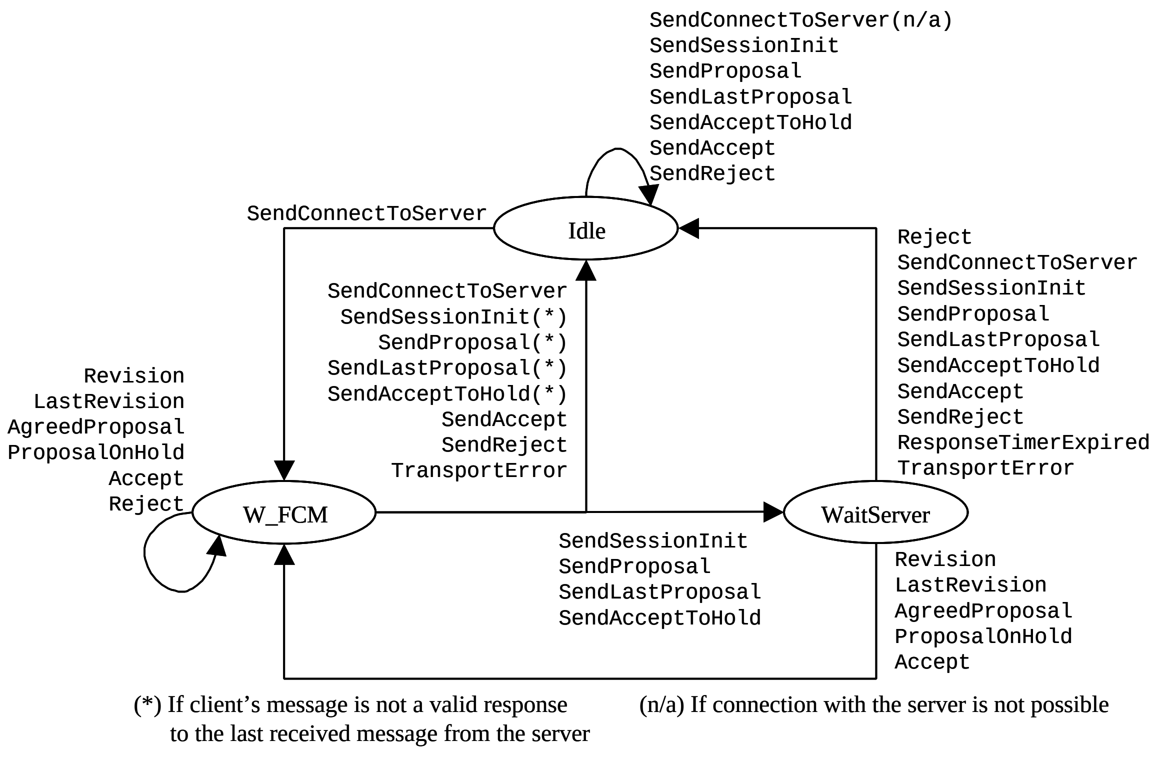


Figure 27. The SrNP client FSM state transition diagram

Client FSM Events	Client FSM States		
	Idle	**W_FCM**	**WaitServer**
SendConnectToServer	ActionConn; W_FCM, Idle	ActionPE; Idle	ActionPE; Idle
SendSessionInit	ActionPE; Idle	ActionSCM; WaitServer, Idle	ActionPE; Idle
SendProposal	ActionPE; Idle	ActionSCM; WaitServer, Idle	ActionPE; Idle
SendLastProposal	ActionPE; Idle	ActionSCM; WaitServer, Idle	ActionPE; Idle
`SendAcceptToHold`	ActionPE; Idle	ActionSCM; WaitServer, Idle	ActionPE; Idle
`SendAccept`	ActionPE; Idle	ActionSCM; Idle	ActionPE; Idle
`SendReject`	ActionPE; Idle	ActionSCM; Idle	ActionPE; Idle
`Revision`	N/A	ActionQSM; W_FCM	ActionRSM; W_FCM
`LastRevision`	N/A	ActionQSM; W_FCM	ActionRSM; W_FCM
`AgreedProposal`	N/A	ActionQSM; W_FCM	ActionRSM; W_FCM
`ProposalOnHold`	N/A	ActionQSM; W_FCM	ActionRSM; W_FCM
`Accept`	N/A	ActionQSM; W_FCM	ActionRSM; W_FCM
`Reject`	N/A	ActionQSM; W_FCM	ActionRSM; Idle
`ResponseTimerExpired`	N/A	N/A	ActionTE; Idle
`TransportError`	N/A	ActionPE; Idle	ActionPE; Idle

Table 5. The SrNP client FSM state transition table

Action	Description
ActionPE	/* Protocol Error */
	Send `ForwardProtocolException` to client's negotiation logic Set next state to Idle
ActionConn	/* To connect to the server to initiate a new negotiation process */ If (connection to the server is possible) Set next state to W_FCM Else ActionPE –reason: connection to server failed
ActionSCM	/* Send to the server the protocol message determined by the client's negotiation logic */ If RejectRegister is not empty –see ActionQSM ActionRSM --to check if any `Reject` message was received from the server while waiting for the client Empty RejectRegister Else If (message to be sent is a valid response from the client to the last received message from the server) Send it to the server If (sent message is `Accept` or `Reject`) Set next state to Idle Else Start `ResponseTimer` Set next state to WaitServer Else Send `Reject` message to the server ActionPE –reason: invalid application behaviour
ActionRSM	/* Receive a protocol message from the server */
	If (received message is a valid and correct response from the server to the last sent client's message) Stop `ResponseTimer` Forward corresponding interface message to the client's negotiation logic If (received message is `Reject`) Set next state to Idle Else Set next state to W_FCM Else Discard received message
ActionQSM	/* Queue server message */ If (received message is `Reject` and a valid and correct response from the server to the last sent client's message) Store received message in the RejectRegister Else Discard received message
ActionTE	/* Response-timer expired */ Send `Reject` message to the server ActionPE –reason: response-timer expired

Table 6. Description of the SrNP client FSM actions

4.3.6.4 SrNP Server FSM

Similarly to the client FSM, the server FSM includes the following states:

- *Idle:* In this state the protocol is (has become) inactive, waiting to be contacted by a new client.
- *WaitToFormServerMessage* (W_FSM): In this state the protocol waits for the server's negotiation logic to determine and formulate its response to the last message received from the client. While at this state, any other message that may come from the client is 'blocked' in the sense that is not forwarded to the server's negotiation logic.
- *WaitClient*: In this state the protocol waits to receive from the client a response to the last sent message determined by the server's negotiation logic. While at this state, the protocol 'blocks' server's negotiation logic in the sense that is not allowed to send to the client any other message that may come from the server's negotiation logic.

The state transition diagram of the server FSM is shown in Figure 28. The associated actions are depicted in Table 7 and described in pseudo-code in Table 8.

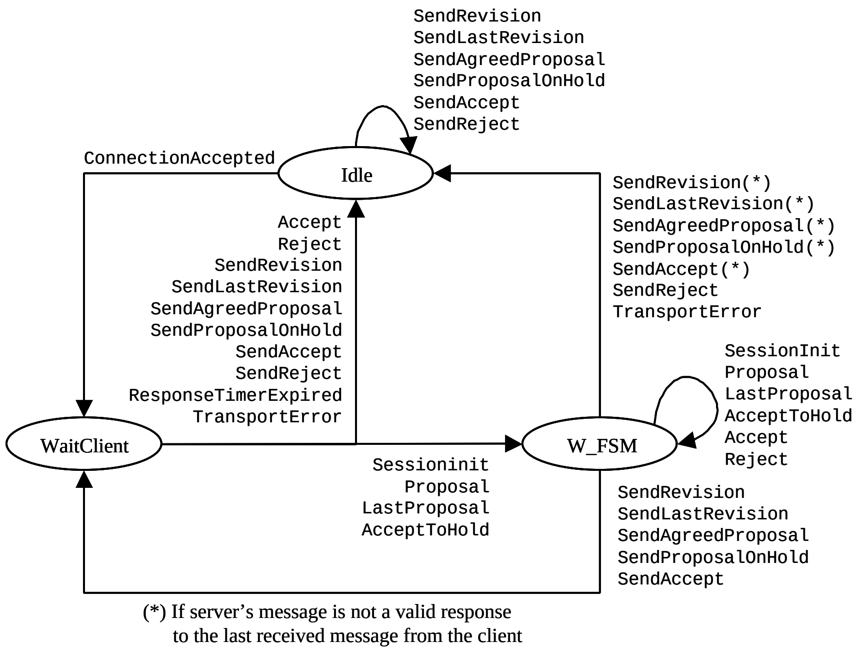


Figure 28. The SrNP server FSM state transition diagram

Server FSM Events		Server FSM States	
	Idle	**WaitClient**	**W_FSM**
`ConnectionAccepted`	ActionConnP; WaitClient	N/A	N/A
`SendRevision`	ActionPE; Idle	ActionPE; Idle	ActionSSM; WaitClient, Idle
`SendLastRevision`	ActionPE; Idle	ActionPE; Idle	ActionSSM; WaitClient, Idle
`SendAgreedProposal`	ActionPE; Idle	ActionPE; Idle	ActionSSM; WaitClient, Idle
`SendProposalOnHold`	ActionPE; Idle	ActionPE; Idle	ActionSSM; WaitClient, Idle
`SendAccept`	ActionPE; Idle	ActionPE; Idle	ActionSSM; WaitClient, Idle
`SendReject`	ActionPE; Idle	ActionPE; Idle	ActionSSM; Idle
`SessionInit`	N/A	ActionRCM; W_FSM	ActionQCM; W_FSM
`Proposal`	N/A	ActionRCM; W_FSM	ActionQCM; W_FSM
`LastProposal`	N/A	ActionRCM; W_FSM	ActionQCM; W_FSM
`AcceptToHold`	N/A	ActionRCM; W_FSM	ActionQCM; W_FSM
`Accept`	N/A	ActionRCM; Idle	ActionQCM; W_FSM
`Reject`	N/A	ActionRCM; Idle	ActionQCM; W_FSM
`ResponseTimerExpired`	N/A	ActionTE; Idle	N/A
`TransportError`	N/A	ActionPE; Idle	ActionPE; Idle

Table 7. The SrNP server FSM state transition table

Action	Description
ActionPE	/* Protocol Error */
	Send `ForwardProtocolException` to server's negotiation logic Set next state to Idle
ActionConnP	/* A client connected to the server */ Start `ResponseTimer` Set next state to WaitClient
ActionSSM	/* Send to the client the protocol message determined by the server's negotiation logic */ If RejectRegister is not empty –see ActionQCM ActionRCM --to check if any `Reject` message was received from the client while waiting for the server Empty RejectRegister Else If (message to be sent is a valid response from the server to the last received message from the client) Send it to the client If (sent message is `Reject`) Set next state to Idle Else Start `ResponseTimer` Set next state to WaitClient Else Send `Reject` message to the client ActionPE –reason: invalid application behaviour
ActionRCM	/* Receive a protocol message from the client */
	If (received message is a valid and correct response from the client to the last sent server's message) Stop `ResponseTimer` Forward corresponding interface message to the server's negotiation logic If (received message is `Accept` or `Reject`) Set next state to Idle Else Set next state to W_FSM Else Discard received message
ActionQCM	/* Queue client message */ If (received message is `Reject` and a valid and correct response from the client to the last sent server's message) Store received message in the RejectRegister Else Discard received message
ActionTE	/* Response-timer expired */ Send `Reject` message to the client ActionPE –reason: response-timer expired

Table 8. Description of the SrNP server FSM actions

4.3.7 Implementation Aspects

4.3.7.1 Protocol Stack

Figure 29 depicts alternative protocol stacks for implementing SrNP. SrNP messages could be encoded in formatted text, BER/TLVs or XML as convenient for the stack used. Note also that it could be possible to encapsulate SrNP messages in widely deployed protocols such as RSVP (by defining new TLVs) and COPS (by specifying a new client-type); prompting, for using SrNP at service invocation times.

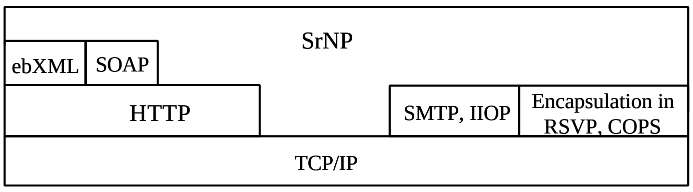


Figure 29. SrNP protocol stacks

4.3.7.2 *Protocol Interface*

SrNP interface to applications realising negotiation logic (cf. Table 4) could be implemented in a number of technologies as desired; different technologies could be used for SrNP client and server interfaces. Typical examples include: APIs in a programming language (e.g. Java, C, C++), IDLs in a Corba-based environment and XML/SOAP in a Web-services environment.

4.3.8 Scalability and Stability Analysis

SrNP scalability is analysed in terms of complexity of its operations, its output -number of protocol messages- and required network resources. The analysis is done in the context of a single negotiation process. It is reminded that SrNP has been designed to apply in negotiation processes involving two parties –one acting in a client role and the other in a server role. Multiple negotiation processes, at the side of a particular negotiating party, are treated independently. Therefore, the complexity of SrNP operations, messages and network resources grows linearly with the number of active negotiation processes, which, after all, is bound by implementation and the capabilities of the party's processing infrastructure. Hence, if SrNP scales in the context of a single negotiation process, so will do in the context of multiple negotiation processes.

As SrNP is dialogue-based, its operations at the side of a particular negotiating party can fall into one of the following two states: waiting for the party's negotiation logic to determine its response to be sent to the other party or waiting to receive a response from the other party (cf. FSMs in Figure 27, Figure 28). As such, SrNP operations involve: in the former state, validity check of the determined response against the last message received from the other party (cf. rightmost column of Table 2), encapsulation into an appropriate protocol message and its forwarding to the other party; and in the latter case, validity and correctness (cf. `messageId`, Table 3) checks of the response received from the other party and its forwarding, after stripping off the protocol header, to the party's negotiation logic. These operations are either atomic or include a look-up into locally available memory structures –for validity and correctness checks. Hence, for a particular negotiation process the complexity of SrNP operations is *O(1)*.

The maximum number of SrNP messages exchanged in the context of a negotiation process is 2+2*Gmin+1, where Gmin=minimum(Gc, Gs) and Gc, Gs is the maximum number of rounds (number of alternate message exchanges) the client, the server respectively is willing to negotiate with the other party. The first term of the sum corresponds to the messages that need to be exchanged for initiating a negotiation process (cf. `SessionInit` client message and `Accept` server message, Table 2) and the last term of the sum corresponds to the ending part of successful negotiations (cf. `Accept` client message, Table 2). SrNP enables negotiating parties to enforce their policy to conclude negotiations within a specific number of negotiation rounds by offering means to: abort negotiations at a given round (cf. `Reject` client and server messages) or force negotiations to conclude, successfully or unsuccessfully, by the next round (cf. `LastProposal` client message and `LastRevision`, `ProposalOnHold` server messages, Table 2).

The network resources required by a particular negotiation process, are those consumed by establishing a single connection at the transport layer, over which SrNP initiates a session for conducting negotiations.

The above discussion proves that SrNP scales.

The stability of SrNP relates to the issue of convergent negotiations i.e. whether SrNP-based negotiations can conclude, successfully or unsuccessfully, in finite rounds and/or time. Because of its very features, as analysed previously, SrNP can terminate in finite rounds. Similarly, SrNP can terminate in finite time. Time restrictions can be enforced by setting accordingly, with respect to the desired number of negotiation rounds, the time period a party is willing to wait for receiving a valid response from the other party (cf. `timeToRespond` field, Table 3). Hence, SrNP converges (can lead to convergent negotiations) provided that the negotiation logic of at least one of the involved parties wishes so –which, can be safely taken for granted.

4.3.9 Conclusions

This Section has presented SrNP, an open protocol for negotiations. In addition to its richness in negotiation primitives, the strengths of the proposed protocol are its simplicity and openness both in terms of technology and separation of concerns from negotiation logic. These strengths make SrNP unique in its kind; currently proposed negotiation protocols are bound to the features of a particular implementation technology or to the negotiation requirements of a specific application domain. By presenting a clear interface to negotiation logic, SrNP offers general-purpose primitives for conducting negotiations, furthermore is agnostic to the semantics and syntax of the issues under negotiation; hence it can apply to a wide range of application domains pertinent to negotiations.

SrNP scales because of the very nature of its design, session-oriented, dialogue-based. In the context of a single negotiation process its operations complexity and required network resources are trivial, while the number of exchanged messages is bound by the minimum of the number of rounds a negotiation party is willing to negotiate with the other party. For multiple negotiation processes, its complexity, required network resources and number of messages grow linearly with the number of active negotiation processes. SrNP is also stable in that it can lead to concluding negotiations in finite steps and/or time, provided that the negotiation logic of at least one of the parties involved wishes so.

4.4 SLS Order Handling

4.4.1 Objectives

The goal of the *SLS Order Handling* function block, as clearly declared by its name, is to handle the orders of the customers of the AS -end consumers and peering ASs- concerning the offered services. The orders come in the form of pSLS, cSLS requests, imprinted using XML documents.

Figure 30 presents the *SLS Order Handling* functional component amongst the other functional components with which it interacts, within the same AS employing the MESCAL functional architecture. In addition the interactions between this AS and its customers communicating with the *SLS Order Handling* component can be viewed.

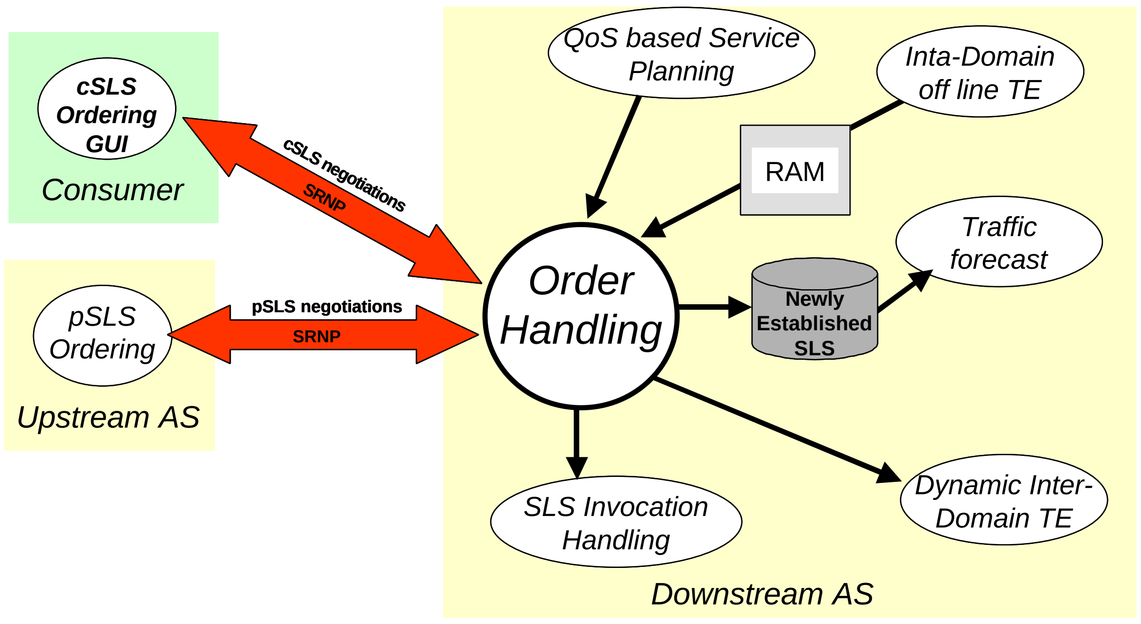


Figure 30. SLS Order Handling

4.4.2 Interface Specification

The *SLS Order Handling* component implements two kinds of interfaces: The external interface for communicating with the peering ASs and the consumers and a set of internal interfaces for communicating with other components of its own AS, as defined by the MESCAL architecture.

4.4.2.1 External Interface

It is based on the service negotiation protocol SrNP, specified by the Tequila project. SrNP realises the communication between the customers of the AS and the *SLS Order Handling* component of the AS by specifying the necessary common vocabulary and setting the negotiation rules that will lead to unambiguous and converging negotiations. In addition it undertakes the transportation of the messages –in form of XML documents – between the negotiating parties.

SLS *Order Handling* implements the server side of SrNP engine, with the capability of accepting several negotiation requests and conducting them simultaneously (Multithreaded).

This interface is expected to be used by the peer ASs requesting the establishment of pSLSs with the AS and by the end customers of the AS requesting the purchase of cSLSs from the AS.

The input from this interface to the *SLS Order Handling* component will be pSLS and cSLS requests in the form of XML documents, adhering to the offered by the AS pSLS, cSLS templates, plus negotiation messages in the form of SrNP messages. The output of this interface will be the response of the *SLS Order Handling* component to the customers in terms of pSLS and cSLS counter-offers and in terms of SrNP messages.

4.4.2.2 Internal Interfaces

According to the MESCAL functional architecture *SLS Order Handling* implements interfaces with:

- *Intra-domain offline TE*. Through this interface *SLS Order Handling* receives as input information the Resource Availability Matrix (RAM), containing the forecasted values of the availability of AS resources both intra and inter domain.
- Business processes. Through this interface *SLS Order Handling* receives policies reflecting the business decisions regarding the Subscription Admission logic.
- *QoS Service Planning*. Through this interface *SLS Order Handling* receives the offered cSLSs including their templates and the accepted bounds of their parameters, also the offered pSLSs and the accepted bounds of their parameters, and finally admission logic policies concerning these pSLSs and cSLSs.
- SLS Repository. Through this interface *SLS Order Handling* stores the established SLSs of the current resource provisioning cycle (RPC) for use of the rest of the components, mainly Traffic forecast.
- *SLS Invocation Handling*. Through this interface *SLS Order Handling* configures the admission logic of the invocation-handling component to take into account and serve accordingly the new subscription.
- *Dynamic Inter-domain TE*. Through this interface *SLS Order Handling* notifies when necessary the dynamic inter domain TE component in the case a qBGP update is needed for the activation of a newly established pSLS or in the case a PCS must be configured with information from new pSLSs or cSLS to be implemented by the hard guarantees solution option.

The interfaces between *SLS Order Handling* and other internal components of the AS, as defined by the MESCAL architecture, do not need any specific protocol for their realisation. They will be based on the exchange of XML documents adhering to mutually agreed templates. The transport of these documents can be undertaken by existing general transport protocols like SOAP or TCP sockets. The documents should be sufficient enough to convey all the necessary input and output information of the *SLS Order Handling* component.

4.4.3 Behaviour Specification

For fulfilling its objectives *SLS Order Handling* should cater for the following functionality:

- *Conduction of negotiations with customers*. The negotiations are initiated by a customer's request and terminated by *SLS Order Handling* accepting/rejecting the service request or the customer rejecting an alternative proposed service. The computational components implementing this functionality are:
    - SrNP Server. It replies to the customers' requests for services and handle the exchange of SrNP messages for the conduction of the negotiations. The server should allow for the simultaneous negotiations with several customers.
    - Translation Engine. It is able to parse the pSLS and cSLS requests, received as XML documents, and deduce the corresponding SLS document representing the network view of the requested connectivity.
    - Authoring Engine. Composes the counter-offer documents, to be proposed to the customer, from the alternative available SLSs deduced from the subscription admission logic.

- *Subscription admission control*. This functionality is responsible for deciding on the faith of the requested SLSs. It provides the logic behind the negotiation process. Taking as input the off-line calculated network availability, the forecasted traffic that will be brought to the network by the requested SLS and policies reflecting planning and business directives, subscription admission control will decide whether to accept or reject the requested SLS. In the case that the SLS is not accepted subscription admission control attempts to calculate alternative SLSs close to the one requested by the customer that can be accommodated by the network. These SLSs are proposed to the customer as counter-offers. The computational components implementing this functionality are:
    - SLS Traffic Forecaster. From the relevant parameters of each requested SLS it deduces the traffic it is expected to burden the network with (based on traffic demand forecast factors). This traffic is then aggregated with the expected traffic accumulated from SLSs established during this resource provisioning cycle (RPC).
    - Admission Logic Enforcer. The resulting aggregated traffic is mapped against the corresponding entries of the resource availability matrix (RAM). The results of this mapping are fed to the admission logic algorithm. This algorithm, configured by appropriate policies, will determine whether the request can be accepted or it must be rejected because the risk of overwhelming the network with traffic that cannot be served with the guaranteed QoS is to high. In case of rejection the upper SLSs traffic limits that the network can gracefully sustain are deduced.
    - Counter-offers Calculator. This component is responsible for calculating the parameters of alternative SLS to the one requested by the customer and rejected. These parameters are deduced based on this rejected SLS and the acceptable SLS traffic limits deduced by the admission logic. These alternative SLSs will be proposed as counter-offers to the customer.
- Establishment of SLSs. This functionality is responsible for triggering the execution of all the necessary actions for activating the newly agreed SLSs, so as for the customer to be able to use the service he has subscribed for. The actions are first the storage of the SLS to a commonly accessible repository and then the notification of the necessary components for the activation of this SLS. These actions are undertaken by the following computational components:
    - *SLS Repository*. It is a central repository accessible by all the components of the management plane and control plane as defined by the MESCAL architecture. Each component of course has certain accessibility rights reflecting its functional needs. In addition it holds all the ASs established SLSs plus additional information like establishment date, the resource provisioning cycle each SLS is accounted for etc.
    - Notification Dispatcher. When an SLS is agreed and stored to the SLSs repository the necessary components for its activation are notified with a message informing them that a new SLS needs activation plus the ID of this SLS –as stored at the SLS repository-. Two components of the control plane undertake the activation of the SLS. The SLS Invocation Handling that is responsible for handling the subscriber’s requests for usage of the service. The Dynamic Inter-Domain TE that is responsible for shipping the necessary qBGP updates and for configuring appropriately the path computational servers (PCS).

4.4.3.1 Subscription Admission Control Algorithm

The goal of subscription admission control, as analysed above, is to determine whether to accept, reject or propose an alternative to a customer’s request for a service – pSLS or cSLS-. The algorithm realising the logic behind the decisions of this component aims at maximising the subscribed traffic without eventually overwhelming the network.

The main input this algorithm will base its decisions on is the resource availability matrix (RAM), which provides an availability estimate per traffic trunk (TT), as calculated by the off-line traffic engineering system (TE). The availability estimate is expressed in the form of an availability buffer per TT. Figure 31 presents this buffer.

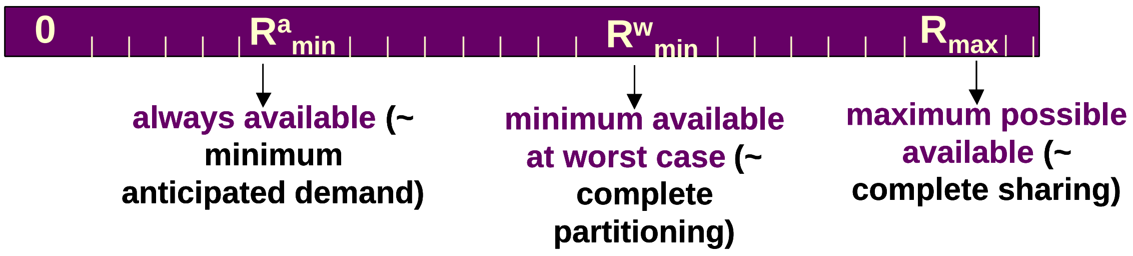

0
R^a_{min}
R^w_{min}
R_{max}
always available (~ minimum anticipated demand)
minimum available at worst case (~ complete partitioning)
maximum possible available (~ complete sharing)

Figure 31. Resource availability buffer

The Ra/min is the first availability limit. It represents the available bandwidth for this TT guaranteed by the network at any time. The Rw/min is the second availability limit. It represents the available bandwidth for this TT guaranteed by the network at congestion times. This bandwidth is not hard reserved by the network and can be utilised by other TTs but if congestion occurs the TE system will force all TTs to be constrained to their Rw/min bandwidth limit. The Rmax is the final limit. It represents the maximum available bandwidth for this TT but with no guarantees because this bandwidth is shared by other TTs.

These availability limits, with the semantics each one bears, are guaranteed by the existing network capacity and the configurations of the traffic engineering system (TE).

In addition to RAM the major identified business policy influencing the behaviour of the admission logic algorithm is the Satisfaction Level (SL) parameter. Figure 32 presents the range of values of the SL parameter along with the semantics of the edge-points of the two main areas this range is divided to.

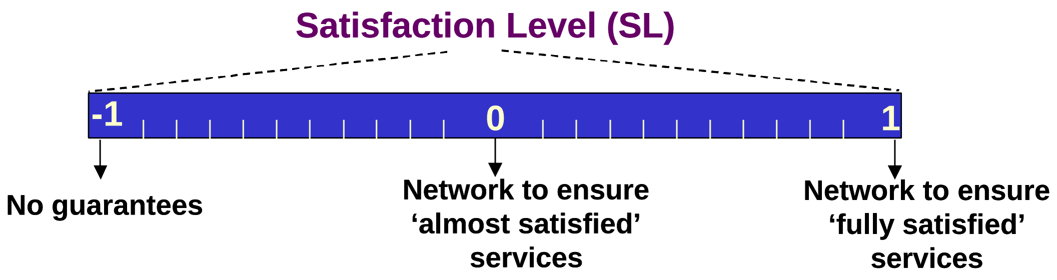

Satisfaction Level (SL)
-1
0
1
No guarantees
Network to ensure 'almost satisfied' services
Network to ensure 'fully satisfied' services

Figure 32. Satisfaction Level (SL), subscription admission control policy

When the value of SL is 1 then the admission logic will accept SLSs up to the point when then network can guarantee that the population of all established SLS will be fully satisfied. By fully satisfied we mean that the SLS will enjoy the upper level of the forecasted traffic per each one of its composing TTs, as calculated by Traffic Forecast (max Demand).

When the value of SL is 0 then the admission logic will accept SLSs up to the point when then network can guarantee that the population of all established SLS will be almost satisfied. By almost satisfied we mean that the SLS will enjoy the lower level of the forecasted traffic per each one of its composing TTs, as calculated by Traffic Forecast (min Demand).

Between 0 and 1, SLS satisfaction will be accordingly between almost and fully satisfied, interpreted that they will enjoy bandwidth between their forecasted min and max demand.

When SL is below 0 the satisfaction of the SLSs will be worst than almost satisfied, that is they will enjoy bandwidth less than their forecasted min demand and if SL decreases up to –1 then the network will not offer any guarantees to the SLSs. At this case the admission logic algorithm will accept all incoming requests for SLS.

The trade off is evident, the more subscriptions you accept, thus increasing your profit, the lower guarantees you offer to your customers for honouring the subscribed SLSs. Clearly it is a business decision whether the company will target many and not so demanding customers or fewer with higher demands and this decision is enforced by appropriately setting the SL policy.

Every time a new service request is addressed at the *Order Handling* component of the AS, as explained before, this request is translated to the corresponding “network view” SLS. The traffic this SLS is expected to burden the network with is deduced, per TT, and aggregated with the forecasted traffic of the established SLSs of this RPC. The result is the anticipated demand that the network will have to serve if the service is accepted. This forecasted demand is expressed by two values the minimum and the maximum demand.

The demand comes as input to the service logic algorithm. The decision, based on the SL and the RAM is made as follows:

- If SL = -1 then accept all SLSs
- If -1<SL<0 then accept the SLSs for witch the portion of the min aggregate demand corresponding to the SL value (the smaller the SL the smaller the portion) is below the Rw/min for each TT.
- If SL = 0 then accept the SLSs for which the min aggregate demand is below the Rw/min for each TT.
- If 0>SL>1 then accept the SLSs for witch the portion of the max aggregate demand above the min demand corresponding to the SL value (the bigger the SL the bigger the portion) is below the Rw/min for each TT.
- If SL = 1 then accept the SLSs for which the max aggregate demand is below the Rw/min for each TT.

If the SLS is not accepted then admission logic should reply with a rejection answer including TTs where the demand of the SLS didn’t fit in the availability bounds as calculated by off-line traffic engineering, plus the value of the maximum demand that could actually accommodated by these TTs. This information is necessary for the calculation of the counter-offers to be proposed as alternatives to the customer.

4.4.4 Test requirements

SLS Order Handling will be tested to demonstrate the capabilities of the implemented functionality when facing various operational requests. More specifically:

- The stability of the process responsible for conducting the negotiations will be tested against all possible negotiation cases.
- The performance and scalability of the process responsible for conducting the negotiations will be tested in cases of increasing simultaneous service requests.
- The functional validity of the translation and authoring processes will be tested against all cases of requested p/c SLSs.
- The performance and scalability of the translation and authoring processes will be tested against all factors influencing them like size of network, population of already established SLS etc.
- The functional validity of the subscription admission control process will be tested in terms of producing the expected results for each case of different inputs-SLS, Satisfaction level, availability matrix-.
- The performance and scalability of the subscription admission control process will be tested against all factors influencing them like size of network, population of already established SLS etc.

4.5 pSLS Ordering

4.5.1 Objectives

The *pSLS Ordering* function block covers all necessary aspects for the completion of the ordering process of pSLSs under specific requests from binding, taking into account imposed negotiation directives.

Figure 33 presents the *pSLS Ordering* functional block and the *Binding Selection* functional block with which it interacts, within the same AS employing the MESCAL functional architecture. In addition the interactions between this AS and its peers realised through the *pSLS Ordering – Order Handling* component communication can be viewed.

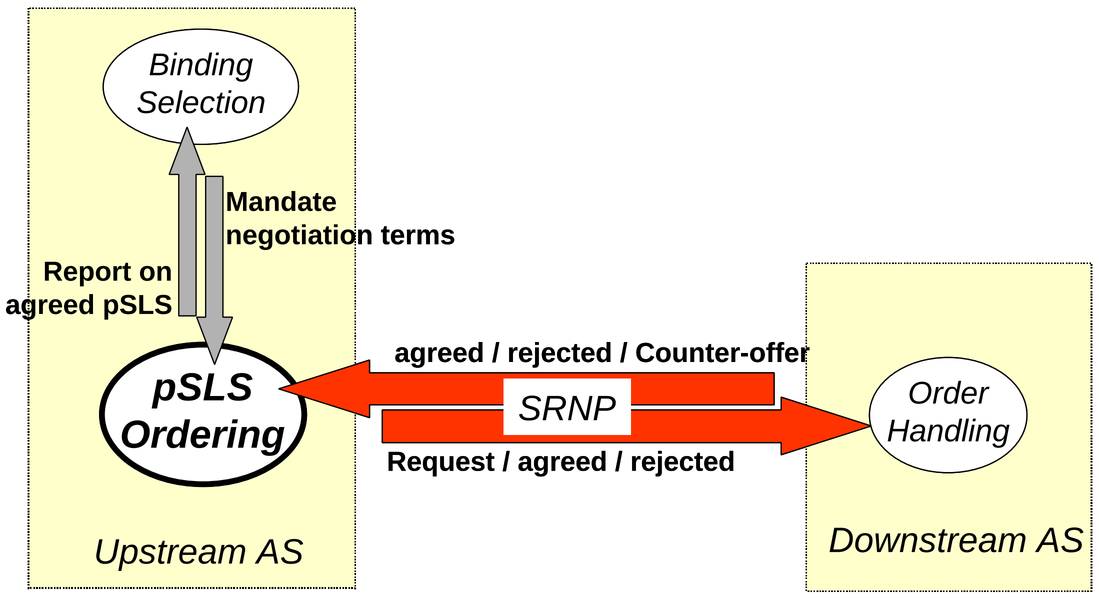


Figure 33. pSLS Ordering

4.5.2 Interface Specifications

The *pSLS Ordering* function block implements two interfaces: The external interface for communicating with the peering ASs and the internal interface for communicating with the Binding Selection component of its own AS, as defined by the MESCAL architecture.

4.5.2.1 External interface with the Order Handling components of peering ASs

The role of the *pSLS Ordering* component is to negotiate and establish pSLSs on behalf of its own AS with the peering ASs. In order for these negotiations to be realised a communications channel needs to be in place between the negotiating parties. The external interface of the *Order Handling* component, as specified above, is implemented as the server side of the SrNP protocol. Therefore the *pSLS Ordering* component must implement the client side of the SrNP protocol.

Acting as an SrNP client *pSLS Ordering* component has the capability to issue requests for pSLSs, in the format of XML documents, to the appropriate AS peers and to receive pSLSs counter-offers, alternative to the requested ones, from them. Also as an SrNP client, *pSLS Ordering* takes part in the exchange of appropriate messages with the server that will conduct the negotiations in an unambiguous and converging manner.

The external interface of *Order Handling* should be able to conduct simultaneous negotiations with multiple peering ASs. To this end several SrNP clients operate at the same time (Multithreaded).

4.5.2.2 Internal interface with the Binding Selection function block

The *pSLS Ordering* component is merely an instrument for negotiating and establishing pSLSs. The decision on the pSLSs, the AS needs to establish and on the terms of their negotiations belongs to the *Binding Selection* function. So *pSLS Ordering* receives input through the internal interface with the *Binding Selection* component concerning:

- Set of pSLSs to negotiate. This set may include already existing pSLSs and request the alteration or even cancellation of them.
- A prioritised list with alternative sets of pSLSs in case the initial set fails to be established.
- For each set, the values and the negotiation margins of aggregated parameters (e.g. total cost).
- For each pSLS within a set, the values and the negotiation margins of its parameters and the AS to negotiate it with.

After the termination of the negotiations, based on the directives of the binding process, *pSLS Ordering* reports the results back to the *Binding Selection*. In the case of success the finally established pSLS including all the agreed parameters are reported, in case of failure the failure is reported along with negotiation logs documenting the reasons of this failure.

The implementation of the internal interface will be based on the exchange of XML documents transported by existing general transport protocols like SOAP.

4.5.3 Behaviour Specification

For realising this goal *pSLS Ordering* should cater for the following functionality:

- *Conduction of negotiations with peers*. The negotiations are initiated by the Ordering component and terminated by either parties when agreement is reached or a final rejection is decided. The computational components implementing this functionality are:
    - Authoring Engine. Composes the XML documents, to be requested by the appropriate peer ASs, corresponding to the pSLSs deduced by the negotiation logic.
    - SrNP Client. It issues pSLS requests and handles the exchange of SrNP messages for the conduction of the negotiations. The simultaneous operation of several SrNP clients should be possible so as to conduct transaction oriented multiple negotiations.
    - Translation Engine. It is able to parse the pSLS counter-offers of the peering ASs, received as XML documents, and deduce the necessary SLS parameters for the negotiation logic.
- *Negotiation logic*. Bears the functionality that realises the logic behind the ordering process. Taking mandates for pSLSs form the binding process tries to reach agreements with the peering ASs satisfying these mandates. This means that it negotiates with the goal of achieving the establishment of the optimum –in terms of cost and performance- pSLSs, always within the given negotiation restrictions. The computational components implementing this functionality are:
    - pSLS Evaluator. At each stage of the negotiations evaluates the current status and determines whether the pSLSs currently under consideration -the proposed ones and the counter-offers received- are within the restrictions given by binding. If this is the case then it requests the establishment of these pSLSs. If not then it requests for a refreshment of the set of pSLSs, under negotiation, from the proposal calculator.
    - Proposal Calculator. Given the current set of pSLSs under negotiation, the stage of negotiations each pSLS is under (e.g. is proposed by the AS, is a counter-offer, is the last proposal etc), the reason this set was rejected by the evaluator and the negotiation restrictions, Proposal Calculator comes up with an alternative set of pSLSs to be negotiated with the peering ASs. This resulting set of pSLSs should be the closer one to

the initial requested set and within the restrictions posed by binding and feasible to be established by the continuation of the current negotiations. If the Proposal Calculator is unable to come up with such a pSLS set then current negotiations have failed. In this case pSLS Ordering initiates the negotiation of the next set of alternative pSLS as set by the binding.

- *Reporting*. After the conclusion of negotiations, Ordering component notifies the *Binding Selection* component on the results. The outcome is reported along with the established pSLSs –including all the finally agreed parameters – in the case of success, or the rejection reasons in case of failure.
    - Report Generator. Undertakes the authoring of the reports reflecting the results of the negotiations and the communication of this report to the *Binding Selection* component.
    - Logs. All the negotiation phases are recorded and stored to logs. These logs can be used for deducing statistics, helpful for various decision making processes of the AS e.g. planning. Also logs are the proof of the negotiations results in case of disputes with the peering ASs.

4.5.3.1 Transaction oriented negotiations logic

pSLS negotiations are handled in a transaction-oriented manner. This means that the negotiation logic doesn't consider pSLSs per case but as a group. The figure below presents the negotiation of four different pSLS between AS1 and its peering ASs bundled together by the negotiation logic to one transactional negotiation.

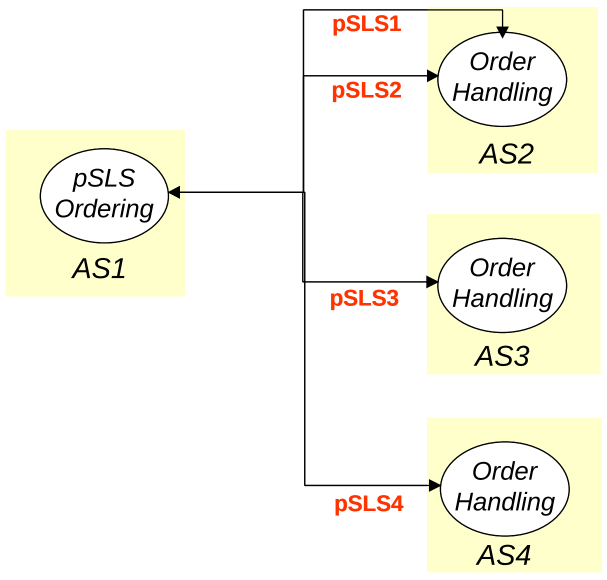


Figure 34. Transaction-oriented pSLS negotiations

Negotiation logic operates based on specific parameters of the SLS. These parameters that can be negotiated are: Destinations, Bandwidth, Traffic Conformance, Grade of Service, Excess treatment, Performance Guarantees, Schedule, Availability Guarantees. The cost is also an important parameter influencing the negotiations not belonging to the specification of the SLS.

All these parameters are negotiated under restrictions and preferences set by the *Binding Selection* function. The restrictions and preferences can be expressed:

- For the value of a single parameter of a single pSLS.
- For the combined values of several parameters of a single pSLS.
- For the aggregate values of a single parameter of all the pSLSs negotiated under the same transaction.

- For the combined aggregate values of several parameters of all the pSLSs of the same transaction.

The negotiation logic process per each transaction should try to achieve pSLS agreements that the values of their parameters will be as close as possible to all the preferences and always within all restrictions.

The negotiation logic operates under the following steps for completing a transactional negotiation:

1. Connect to all involved –to this transactional negotiation- peering ASs
2. Simultaneously propose them the first set of pSLSs demanded by binding selection with parameter values equal to the preferred values.
3. Receive all responses
4. If all the proposed pSLSs are accepted then establish them. The negotiation is terminated with success.
5. If we have responses other than acceptations (rejections, counter-offers) evaluate the set of pSLS arising from these responses against the posed negotiation restrictions.
6. If the set under consideration is evaluated as accepted then establish all proposed pSLSs. The negotiation is terminated with success.
7. If not then calculate a new set of pSLSs, with parameter values within the restrictions and as close as possible to the preferred values, that will be used as the new proposal for the continuation of the negotiations.
8. If such a new proposal is feasible to be formed continue negotiations from step 2 with it.
9. If not, this transactional negotiation has failed and rejection messages must be send to all involved peering ASs.

4.5.4 Test requirements

pSLS Ordering will be tested to demonstrate the capabilities of the implemented functionality when facing various operational requests. More specifically:

- The stability of the process responsible for conducting the transactional negotiations will be tested against all possible negotiation cases.
- The performance of the process responsible for conducting the transactional negotiations will be tested in cases of increasing size of pSLS sets and of increasing complexity of negotiation messages.
- The functional validity of the translation and authoring processes will be tested against all cases of requested pSLSs.
- The performance and scalability of the translation and authoring processes will be tested against all factors influencing them like size of network, pSLS complexity etc.
- The functional validity of the transaction oriented negotiation logic process will be tested in terms of producing the expected results for each case of different inputs- pSLS sets, negotiation restrictions, providers' counter-offers -.
- The performance and scalability of the transaction oriented negotiation logic process will be tested against all factors influencing them like of increasing size of pSLS sets and of increasing complexity of negotiation restrictions.

4.6 cSLS Ordering

4.6.1 Objectives

Figure 35 presents the realisation of the *cSLS Ordering* functionality. The purchase of cSLSs is achieved by the consumers through a simple web-browser that communicates with the provider's cSLS Offering web-server. This server publishes web pages, according to the provider's cSLS offerings as decided by QoS Based Service Planning and acts as the mediator for the cSLS negotiations between the consumer and the Order Handling component.

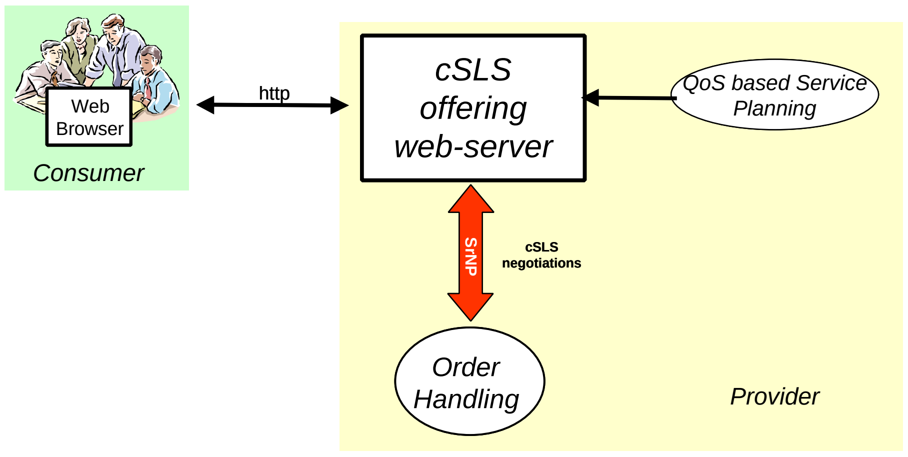


Figure 35. cSLS Ordering

4.6.2 Interface Specification

cSLS Offering server implements three interfaces: an external with the consumers and two internal ones with the *QoS-Based Service Planning* and with the *SLS Order Handling* components of the provider. Through these interfaces the offering of cSLSs is automated and the subscription process is simplified for the end consumer.

4.6.2.1 External interface with the end consumers

This is an interactive web interface accessible by the consumers through their web browser. It offers to the consumers:

- A catalogue of the offered cSLSs along with their detailed descriptions.
- Templates through which the consumers can issue requests for cSLS according to their needs.
- Actions for conducting the negotiations.
- The responses of the provider to the consumers' requests and negotiation actions

4.6.2.2 Internal interface with the QoS Based Service Planning component

Through this interface *cSLS Offering* Server receives the provider's offered cSLSs including their description, templates, presentation directives and offering rules. Based on this input the *cSLS Offering* Server constructs the dynamic web pages implementing the external interface with the consumers.

4.6.2.3 Internal interface with the Order Handling component

cSLS Offering Server negotiates the establishment of cSLSs on behalf of the customers. Therefore it needs to communicate with the *SLS Order Handling* component, which is responsible for negotiating the offering of the cSLS on behalf of the provider. The *SLS Order Handling* component as specified above implements the server side of SrNP so the Offering Server must implement the client side of SrNP.

All the necessary communications for the realisation of the negotiations, as described in the case of the *pSLS Ordering – SLS Order Handling* interface, are supported by this interface. The conduction of simultaneous negotiations on behalf of several consumers is again a requirement.

4.6.3 Behaviour Specification

The *cSLS Offering* Server realises the mediation part of the *cSLS Ordering* process by fulfilling the communication between the consumers and the provider. This is achieved by implementing the above-specified interfaces. The logic behind the ordering process belongs solely to the consumer. The functionality that needs to be employed by this component is:

- *Publication of cSLS offerings*. This functionality must cater for presenting the offered cSLSs to the consumers, as decided by the planning of the provider. The computational components implementing this functionality are:
  - Web Page Factory. On receiving the new cSLSs or alteration/cancellation demands for old ones it produces automatically the necessary web pages, based on directives from *QoS Based Service Planning*, for presenting these cSLSs and enabling their ordering by the consumers.
  - Dynamic Web Site. This is the web site that contains all the pages dynamically created by the Web Page Factory corresponding to all the offered cSLSs. In addition it offers sign in, connection to service billing process and transaction logging capabilities.
- *Mediation between consumers - Order Handling.* This functionality must cater for transporting the consumers' requests to the *SLS Order Handling* component and visualising the resulting responses via appropriate web pages. The computational components implementing this functionality are:
  - Message Translation Engine. It translates the consumers requests and negotiation demands to the corresponding SrNP messages and transforms the SrNP responses form the *SLS Order Handling* component to the appropriate web pages for the continuation of the negotiations.
  - SrNP Client. It realises the communication with the SrNP server of the *SLS Order Handling* component. The simultaneous operation of several SrNP clients should be possible so as to serve several consumers requests at the same time.

4.6.4 Test requirements

cSLS Offering server will be tested to demonstrate the capabilities of the implemented functionality when facing various operational requests. More specifically:

- The functional validity of the Web-Page Factory operation will be tested in terms of producing the expected results for each case of different cSLS offerings decided by QoS service planning.
- The functional validity, performance and scalability of the Dynamic Web-Site against increasing size and diversity of consumers' requests.
- The functional validity, performance and scalability of the Mediation process against increasing size and diversity of consumers' requests and increasing complexity of *SLS Order Handling* responses.

4.7 pSLS Invocation

4.7.1 Objectives

The *pSLS Invocation* function block is an offline component that is responsible for invoking pSLSs with peer domains. The pSLSs have already been subscribed through an ordering process between *pSLS Ordering* and *SLS Order Handling*. The ordering process establishes the overall boundaries and performance guarantees of the transport agreement between the peering domains, but an invocation process is required before user traffic can be passed between the domains. The invocation may request that the entire bandwidth that has been ordered is committed or it may request that only a portion is committed. Invocations may occur at intervals that are shorter than the Resource Provisioning Cycle epoch.

4.7.2 Interface specification

This section describes the interaction of the pSLS Invocation functional block with other functional blocks, as specified in the MESCAL functional architecture [D1.1]. Figure 36 shows the interfaces related with SLS Invocation.

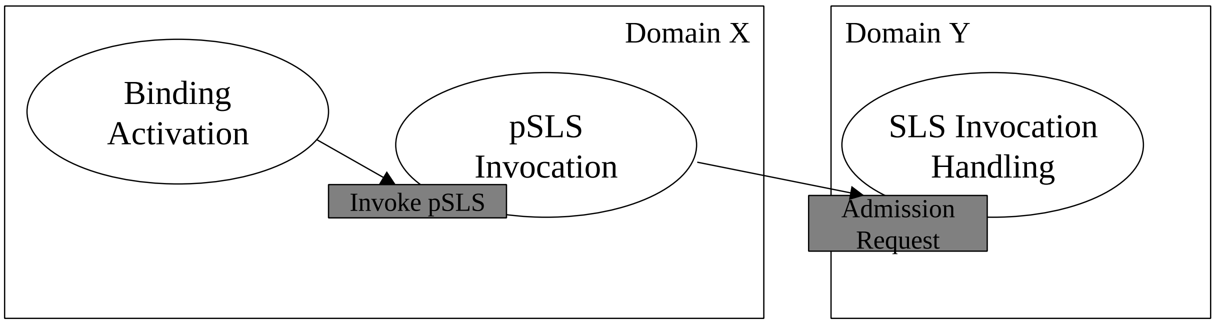


Figure 36. pSLS Invocation

- **Binding Activation to pSLS Invocation**

Invoke pSLS (eRAM)

Binding Activation determines the optimum arrangement of Inter-domain pSLSs to satisfy the TE requirements for a particular time interval and passes the information to pSLS Invocation. The eRAM identifies the peering domain, by specifying an egress interface, for each pSLS along with the required bandwidth.

- **pSLS Invocation to SLS Invocation Handling**

Admission Request

Each pSLS Invocation requires a request to be forwarded to the *SLS Invocation Handling* process in the peering domain, which is responsible for performing admission control.

4.7.3 Behavioural specification

Binding Activation determines the Inter-domain TE solution, as a set of pSLSs, that need to be established with peering domains. The pSLS requirements are conveyed as a set of entries in an eRAM. The eRAM contains the necessary information to enable pSLS Invocation to invoke the pSLSs, although it may be supplemented with pSLS identifiers, which are known from the prior pSLS subscription/handling process.

pSLS Invocation is responsible for identifying the appropriate peer domain and requesting that the peer domain admit the pSLS. If successful, *Binding Activation* is informed and the domain is prepared to transport QoS traffic over this path. If unsuccessful, *Binding Activation* is informed and it is *Binding Activation's* responsibility to decide on the next action.

It is possible that a set of pSLSs may have to be regarded as atomic by pSLS Invocation, which means that all invocations in the set must be invoked if the action is to be considered successful.

The negotiation between *pSLS Invocation* and *SLS Invocation Hand*ling can be based on the SrNP protocol described previously, although a simplified version would be sufficient as invocation requires less complex negotiation than the subscription/handling process.

4.7.4 Test Requirements

It is not intended to develop a protocol-based implementation this component in the implementation/experimental phase of the project and therefore no specific test requirements are specified. *pSLS Invocation* is of course required to enable evaluation of inter-domain QoS but this will be achieved by ad hoc means that do not require the use of a protocol-based solution.

4.8 SLS Invocation Handling

SLS Invocation Handling (Admission Control) is an online component that is responsible for controlling the amount of traffic injected into the network so that conformant users achieve predefined performance objectives, as these are specified in the (c/p)SLSs. The term 'users' can correspond to either individual customers, e.g. home users, universities, organisations, or to entire provider domains. In the former case, cSLSs describe the required performance guarantees, whereas in the latter case, pSLSs describe the required traffic treatment. *SLS Invocation Handling* is a necessity in QoS-enabled networks and should act proactively so as to prevent saturation of the available resources and its consequences before they actually happen. Depending on the QoS guarantees that need to be provided and other factors, such as type (real-time or elastic) and aggregation level of carried traffic (cSLSs or pSLSs), overbooking ratios etc, different QCs will require different admission control policies and algorithms. Admission control needs also to take into account potential statistical multiplexing gain and interactions between the QCs that share the same links. Additionally the type of QoS parameters declared in the SLSs (e.g. peak rate only or other traffic descriptors) will greatly influence the employed admission control schemes.

4.8.1 Objectives

The main objective of *SLS Invocation Handling* is to guarantee that, once admitted, customer/domain service requests will receive the pre-agreed QoS treatment with the agreed guarantees for their entire duration, as these are described in the corresponding SLSs, without causing any downgrade to the already established services. An additional objective of *SLS Invocation Handling* is to optimise the use of network resources.

Note that *SLS Invocation Handling* must be capable of achieving these two objectives under any offered traffic conditions in the context of the statistical and hard guarantees solution option. Loose SLS invocation handling is required for the loose guarantees solution option.

4.8.2 Interface specification

This section describes the interaction of the *SLS Invocation Handling* functional block with other functional blocks, as specified in the MESCAL functional architecture [D1.1], through events, messages or signals. Figure 37 shows the interfaces related with *SLS Invocation Handling*.

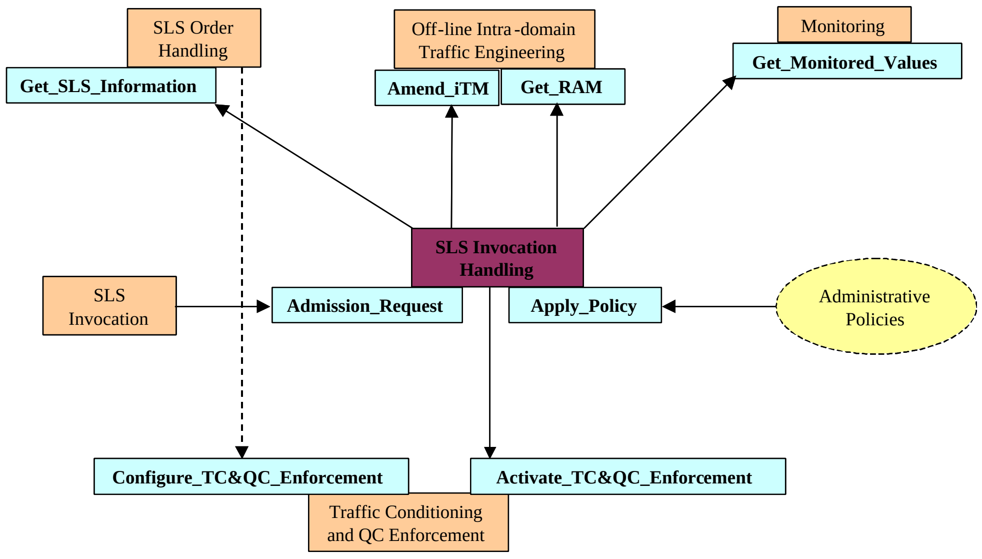


Figure 37. The SLS Invocation Handling interfaces.

- **SLS Invocation to SLS Invocation Handling**

  Admission_Request

  This method will be called by *SLS Invocation* to initiate the *SLS Invocation Handling* process.

- **SLS Order Handling to SLS Invocation Handling**

  Get_SLS_Information

  This method will be called by *SLS Invocation Handling* to request and get SLS information from the *SLS Order Handling* functional block regarding the number and type of the subscribed SLSs. The SLS subscription process should take into account the available resources, as these are expressed in the Resource Availability Matrix (RAM), outputted by the *Off-line Intra-domain Traffic Engineering* functional block.

- **SLS Order Handling to Traffic Conditioning and QC Enforcement**

  Configure_TC&QC_Enforcement

  This method will be called by *SLS Order Handling* to configure the traffic conditioners and bind the l-QCs that will carry the traffic of the subscribed SLSs with the appropriate l-QC/o-QCs of the peering domain, if traffic needs to be carried to destinations that cannot be reached within the domain.

- **Off-line Intra-domain Traffic Engineering to SLS Invocation Handling**

  Get_RAM

  This method will be called by *SLS Invocation Handling* to request and get the Resource Availability Matrix (RAM) from *Off-line Intra-Domain Traffic Engineering*. RAM, which is derived from the internal RAM (iRAM) and the external RAM (eRAM) provides estimates of the available resources end-to-end for all employed QCs or Meta-QoS-Classes. The estimates are provided as a range of values allowing for potential resource sharing among the employed QCs or Meta-QoS-Classes.

- **Monitoring to SLS Invocation Handling**

  Get_Monitored_Values

  This method will be called by *SLS Invocation Handling* to request and get information from *Monitoring* regarding the actual state of the network by means of real-time measurements. The measured entities can be bandwidth, one-way packet delay, packet delay variation and packet loss rate.

- **Administrative Policies to SLS Invocation Handling**

  Apply_Policy

  This method will be called by the Administrative Policies block to apply predecided policies that will influence the *SLS Invocation Handling* decision-making process.

- **SLS Invocation Handling to Traffic Conditioning and QC Enforcement**

  Activate_TC&QC_Enforcement

  This method will be called by *SLS Invocation Handling* to trigger the activation of the appropriate *Traffic Conditioning and QC Enforcement* following every successful SLS invocation request.

- **SLS Invocation Handling to Off-line Intra-domain Traffic Engineering**

  Amend_iTM

  This method will be called by *SLS Invocation Handling* to trigger the recalculation of the internal Traffic Matrix (iTM), in case of excessive subscribed SLSs invocation rejections. This situation could occur when the available resources are oversubscribed, or the demands of the subscribed SLSs, in terms of bandwidth, are underestimated.

4.8.3 Behavioural specification

SLS Invocation Handling is performed at the network ingress routers and is responsible for accepting/rejecting SLS invocation requests on the behalf of an entire network domain or a sequence of domains, if the destination address of the traffic, points to another domain. The SLS Invocation process can be initiated either explicitly, e.g. through RSVP signalling, or implicitly. Upon receipt of an SLS Invocation request, the *SLS Invocation Handling* block will use the following inputs and derive the following outputs, as these are depicted in Figure 38.

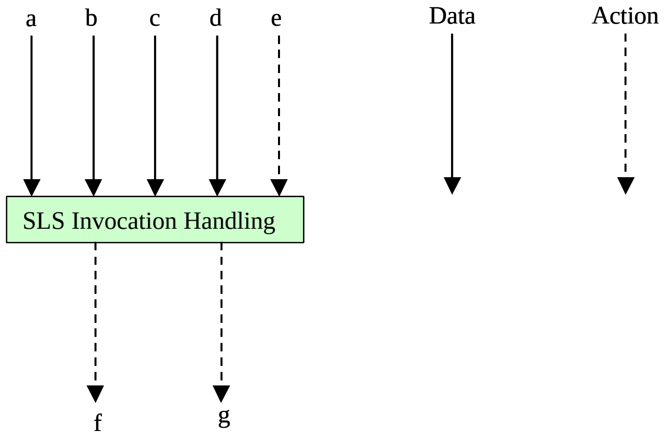


Figure 38. SLS Invocation Handling Inputs/Outputs.

4.8.3.1 Inputs/Outputs

<u>Input data:</u>

a. *SLS information*

SLSs may include the following:

Parameter Group	Description
Customer/user identifier	Identifies the customer or the user for Authentication, Authorisation and Accounting (AAA)
Flow descriptor	Identifies *the packet stream* of the contract by e.g. specifying a packet filter (DSCP, IP source address, etc).
Service Scope	Identifies the geographical region *where* the contract is applicable by e.g. specifying ingress and egress interfaces.
Service Schedule	Specifies *when* the contract is applicable by giving e.g. hours of the day, month, year
Traffic descriptor	Describes the traffic envelope through e.g. a token bucket, allowing identification of in- and out-of-profile packets
QoS Parameters	Specifies the QoS network guarantees offered by the network to the customer for in-profile packets including delay, jitter, packet loss and throughput guarantees.
Excess Treatment	Specifies the treatment of the out-of-profile packets at the network ingress edge including dropping, shaping and re-marking.

b. RAM

RAM should give an estimate of the available resources for the various reachable destination prefixes (end-to-end) for all employed QCs or Meta-QoS-Classes. The estimates could be expressed as a range of bandwidth values, taking into account potential resource sharing between classes and intra/inter-domain reachable destination prefixes.

c. Monitored values

Monitored values are needed for measurement-based *SLS Invocation Handling* approaches. They depict the network condition in real-time and can involve bandwidth, packet loss, delay and delay variation measurements.

d. Number and type of already invoked SLSs

Model-based *SLS Invocation Handling* processes require knowledge of the number and type of established services at all times. This means that services must signal, apart from their initiation, their termination. If this is not feasible, an alternative is to employ a time-out period, based on the activity of a service, as an indication of its termination.

Input Actions:

e. Policies

Predefined policies that influence the *SLS Invocation Handling* decision. Policies can determine facts, such as the level of conservativeness the administrator is willing to enforce with respect to the satisfaction of the predefined performance objectives, the preferential treatment of some applications (e.g. real-time) compared to others, and the treatment of invocations that don't correspond to subscribed services.

Output actions:

f. Traffic Conditioning and QC enforcement

Upon an SLS Invocation admission, at ingress routers, actions such as packet classification, policing, shaping and DSCP marking, according to the conditions laid out in the previously agreed SLSs, are taken. QC enforcement is responsible for implementing the binding of the employed l-QC to the service peer o-QC, in case the destination prefix of the service requires inter-domain transition.

g. Traffic Matrix Amendment

Excessive service rejection rates may indicate oversubscription of the resources or underestimation of the bandwidth requirements of the subscribed SLSs. In all cases, an iTM recalculation might be needed for the next Resource Provisioning Cycle.

4.8.3.2 Process Description

Upon an SLS Invocation request, *SLS Invocation Handling* will use SLS information to check whether the initiating user, either customer (cSLS) or domain (pSLS), is authorised to request the specific service. If there exists a subscription for the requested service, the SLS Invocation request will always be considered for admission. If there does not exist a subscription for the requested service, then policies will determine whether the request will be further considered or immediately dropped. The applied policies will take into account facts, such as the type of traffic the service request will send and the current state of the network (e.g. if the available resources for the QC that will be employed for carrying the traffic of the requested service are more than a threshold, then consider, otherwise reject) If a request is decided to be considered for admission, through monitoring (measurement-based approach) and/or by using traffic descriptors (model-based approach), and by taking into account the information of the RAM, the available resources for the o-QC or Meta-QoS-Class that will be used to carry the traffic injected by the service are estimated. Another approach for indirectly estimating the available resources is by sending a stream of probing packets end-to-end (endpoint approach). If the resources are adequate to support the service, it is admitted and the appropriate *Traffic Conditioning and QC enforcement* actions are triggered. If the resources are not adequate, the service request is rejected. Note, that policy reasons may require a service request to be rejected even if the available resources are adequate. The service rejection rate, especially of requests corresponding to subscribed SLSs, needs to be maintained, since excessive values may indicate traffic engineering and oversubscription problems.

4.8.4 SLS Invocation handling issues

4.8.4.1 Case Studies

SLS Invocation Handling will need to consider and discriminate the following options, and different approaches and algorithms might be required for each possible case:

- Real-time traffic vs. Elastic traffic

  Real-time (UDP controlled) and elastic traffic (TCP controlled) have different QoS requirements and exhibit different traffic patterns. Furthermore, the invocation of elastic traffic flows is mainly implicit (HTTP traffic) whereas for real-time traffic is explicit. Therefore, an SLS invocation handling approach that suits one type of traffic may not achieve satisfactory performance for the other type of traffic.

- Peak rate allocation vs. Statistical multiplexing allocation

  In the first approach, for bandwidth allocation, each SLS is allocated bandwidth equal to its declared peak rate whereas in the second approach we allow for resource sharing between the SLSs belonging to the same o-QC/Meta-QoS-Class. The first approach can provide harder QoS guarantees but can lead to poor network utilisation.

- cSLSs vs. pSLSs

  cSLSs and pSLSs may differ greatly regarding the aggregation level and the characteristics of traffic. Therefore, an SLS invocation handling approach that suits one type of SLSs may not achieve satisfactory performance for the other type of SLSs.

- Interactions between classes

  SLS Invocation Handling needs to take into account potential interactions between different classes in terms of resource sharing and relative priority in terms of scheduling.

- Overbooking vs. Non-Overbooking

  The two cases may require different treatment depending on the employed SLS invocation handling approach for providing the same QoS guarantees for the invoked and admitted service requests.

- Endpoint vs. Measurement-based vs. Traffic descriptor-based approaches for determining available resources

  The employed *SLS Invocation Handling* algorithms will be greatly determined by the approach used to estimate the available resources. For the endpoint approach, the estimation is based on the calculation of some metrics on streams of probing packets. For the measurement-based approach, the estimation of resources is based on real-time measurements of the actual network traffic, whereas for traffic descriptor-based approaches the estimation relies totally on the declared traffic descriptors included in the SLSs.

- Bi-directionality issues

  SLS Invocation Handling for bi-directional services, such as Voice-over-IP will need to take into account not only the state of the forward path, that is the end-to-end path that will be used by the SLS originated traffic, but additionally the state of the return path, that is the end-to-end path for the receiver initiated traffic.

4.8.4.2 *Examples*

With respect to the aforementioned cases, the *SLS Invocation Handling* approaches as presented in [D1.4] and [SGPT04] address the following:

In [D1.4]:

- Real-time traffic
- Peak rate allocations
- cSLSs
- No interactions between classes
- Overbooking
- Measurement-based approach (based on congestion indications –green/red states)
- Bi-directionality not taken into account

In [SGPT04]:

- Real-time traffic
- Statistical multiplexing allocations
- cSLSs
- No interactions between classes
- Overbooking
- Combined Measurement and Traffic Descriptor-based approach
- Bi-directionality not taken into account

4.8.5 SLS Invocation Handling Algorithms

In general we can separate the SLS traffic based on its responsiveness to congestion. The categorisation, which is widely accepted today, is that between elastic and non-elastic traffic. The latter is also known as real-time traffic. Examples of elastic traffic sources are sources that use TCP as the transport protocol, while real-time traffic sources are applications that use UDP without any congestion control at the application level.

In the following we will look into algorithms for admission control of real-time and elastic traffic sources.

4.8.5.1 *Real-time Traffic Admission Control*

In this section we consider admission control for real-time traffic. We define as real-time traffic sources, the ones which have a strict small delay requirement and a bounded, not necessarily too low, packet drop rate (PDR) requirement [BON01]. In a Diffserv domain, the PHB used for this traffic will be the Expedited Forwarding (EF). We assume that such traffic will be aggregated to form one or more real-time traffic aggregates, and that the traffic from the sources that composes each traffic aggregate will receive the same treatment over the entire domain. The delay requirement of the traffic aggregate has been taken into account in the provisioning stage, i.e. by appropriately setting small queues and by manipulating the routing process to choose appropriate paths. Packets are expected to be significantly queued and lost only at the first point of aggregation (ingress node), where we are going to have the serialization [BON01] of the various traffic sources. We assume that the interior of the Diffserv domain has been provisioned and engineered in order to support the real-time traffic aggregates. That implies that further downstream, traffic aggregates are treated in a peak rate manner. This is feasible since, as stated in [MAY01], in a common network configuration, backbone links are over-provisioned.

Low jitter is also a requirement for real-time traffic, but according to [BON01], if the network deploys the EF PHB using priority queuing and certain conditions are met, the jitter of the individual sources remains negligible. These conditions are that either the probability of the combined input rate of individual sources exceeds the service rate with a sufficiently small probability, which is inline with the real-time traffic requirements, or the sum of input rates does not exceed the service rate. Regarding our approach, at the first aggregation point the first condition is met, whereas further downstream the second condition is met.

The support of the real-time traffic aggregate requires engineering and provisioning decisions, in terms of capacity and routing management, in order to achieve the QoS requirements, (delay, loss), of the individual sources. As a result of this provisioning process, and taking into account the routing behaviour, at each ingress node we have an estimate of the minimum available bandwidth that is available from that ingress to each of the corresponding egress nodes. This available bandwidth is the basis for our admission control approach, which is employed at the edge (ingress) node of the first Diffserv aggregation point, for accepting a traffic source on behalf of the entire network domain or a sequence of domains, if the destination address of the traffic, points to another domain.

4.8.5.1.1 SLS Invocation Handling Framework

In Diffserv networks, customer traffic contracts are pre-agreed through SLSs [MYK03] that include traffic descriptors. Our proposed approach is a combination of Measurement-based and *a priori* Traffic descriptor Admission Control -we will be referring to it as MTAC. As such, it encompasses the positive features of both admission control schemes. It provides a systematic way to derive the bandwidth requirements of the already established flows through measurements and of the candidate for admission flow through its traffic descriptors. We use real-time measurements of the actual load in order to cope with the fact that the traffic descriptors may not depict the actual characteristics of individual flows. In this case, we might end up with either under-utilisation, if users overestimate their requirements, or with QoS degradation if users underestimate their requirements and the network is not able to police them efficiently. For bandwidth manipulation and allocation we have adopted the effective bandwidth approach. According to [GUER91], when the effect of statistical multiplexing is significant, the distribution of the stationary bit rate can be accurately approximated by a Gaussian distribution. Regarding the validity of this assumption, in [SHRO03] it is strongly suggested that the aggregation of even a fairly small number of traffic streams is usually sufficient for the Gaussian characterisation of the input process. In [GUN92], it is suggested that a number of aggregated sources as low as ten is enough for the Gaussian assumption to hold. In that case, the effective bandwidth of the multiplexed sources is given by:

$$C \approx m + a's \text{ with } a' = \sqrt{-2\ln(e) - \ln(2p)} \qquad (1)$$

where m is the mean aggregate bit rate, s is the standard deviation of the aggregate bit rate and e is the upper bound on allowed queue overflow probability.

4.8.5.1.1.1 SLS Invocation Handling Logic

We assume that through provisioning and traffic engineering, C_{total} bandwidth is available edge-to-edge for the real-time traffic aggregate. We use a *reference source*, with mean and standard deviation (m_{ref}, s_{ref}) as a model source for engineering reasons. We define as *reference trunks* (T_{ref}) the number of simultaneously established reference sources that can fit in C_{total} for a given target bound on packet drop rate. In our model, we assume that every time a source wants to establish or terminate a service instance, it signals this to the ingress node through a resource reservation protocol. As such, the number of active sources at every point in time is always known.

When a new request arrives, measurements are taken for bandwidth estimations. The measured parameters are the mean rate of the offered load, $M_{measured}$, and the variance of the offered load, $s^2_{measured}$, at the output queue of the ingress node. We then calculate the number N_m of the reference sources, whose aggregate mean rate is equal to or greater than $M_{measured}$, and the number N_s of the reference sources, whose aggregate variance is equal to or greater than $s^2_{measured}$. That is, N_m and N_s are the integers that satisfy the following relationships:

$$N_m = \left\lceil \frac{M_{measured}}{m_{ref}} \right\rceil \text{ and } N_s = \left\lceil \frac{s^2_{measured}}{s^2_{ref}} \right\rceil \qquad (2)$$

Having estimated N_m and N_s , we compute their mean value N_{ref}, which represents an estimate of the number of reference sources that produce load with characteristics similar to the ones having been measured.

$$N_{ref} = (N_m + N_s)/2 \qquad (3)$$

The new source requesting admission has declared in the SLS its traffic descriptors (m_{new}, s_{new}). In case the only available traffic descriptor is the new source's peak rate p_{new}, the above pair becomes $(m_{new}, s_{new}) = (p_{new}, 0)$. In all cases, having the measurements and the traffic descriptors of the new source, we compute the estimated bandwidth C_{est} as follows:

$$C_{est} = M_{measured} + m_{new} + a'_{PDR}\sqrt{s^2_{measured} + s^2_{new}} \qquad (4)$$

where a'_{PDR} is computed as in (1), based on the target PDR bound of the real-time traffic aggregate. This value C_{est} will be used in the admission control criterion.

4.8.5.1.1.2 The Precaution Factor (PF)

Before deriving the admission control criterion, there are two important issues that need to be taken into account. The first one is that the admission control decision needs to be more conservative as the current measurements of the offered load correspond to a number of reference sources N_{ref} that exceeds the number of reference trunks T_{ref} to avoid degradation of performance due to excessive PDR incurred by the overloading of the system.

The second important issue is that the more stringent the target bound on PDR, the more conservative the admission control decision should be. To demonstrate why that should hold, one can consider the following example: the bandwidth that is required by 100 VoIP sources, with peak rate 64kbps and exponentially distributed ON and OFF periods with average durations 1.004sec and 1.587sec, with a bound on queue overflow probability equal to 0.01, using (1), is 3.22608 Mbps. According to (1), the bandwidth required for 101 sources of that type for queue overflow probability 0.01 is 3.355630 Mbps. That means that the additional source gives an increase of bandwidth, $\Delta_{eff}(0.01)$ = 29.022kbps. For queue overflow probability 0.001, the corresponding increase is $\Delta_{eff}(0.001)$ = 30.182kbps. This means that since for decreasing bound on queue overflow probability (which directly translates to a decreasing bound on PDR) the increase in bandwidth requirements for each admitted source is greater, it should be taken into account in the admission control decision.

When we decide whether a new source should be admitted or not, the two issues discussed above need to be taken into account. Therefore we introduce a *Precaution Factor* (PF) before we derive the final expression for our admission control criterion. The more conservative the admission control decision, the greater the value of the precaution factor.

In order to take into account the first issue, PF should be proportional to the quantity (N_{ref}/T_{ref}). In order to take into account the second issue, we proceed as follows: given (1), for two different levels of PDR, e.g. e_1 and e_2 with $e_1 < e_2$ and all other parameters (number and characteristics of sources) the same, it can be suggested that this relative increase in additional bandwidth required is proportional to the quantity $\frac{\sqrt{-2\ln(e_1)-\ln(2p)}}{\sqrt{-2\ln(e_2)-\ln(2p)}}$.

Therefore, the final expression for the precaution factor that is adopted is:

$$PF = (N_{ref} / T_{ref}) * \frac{\sqrt{-2\ln(e)-\ln(2p)}}{\sqrt{-2\ln(e_{ref})-\ln(2p)}} \qquad (5)$$

In both expressions, e_{ref} is a reference PDR level. Also, when it is $N_{ref} < T_{ref}$ then the value $PF = 1$ is adopted.

4.8.5.1.1.3 *The Measurement Window*

We define the measurement window w, as the time interval within which the offered load is taken into account for deriving the required measurements. In similar fashion to [BELE02], we use the following expression for the measurement window:

$$w = \max(DTS, w'), \qquad (6)$$

In (6), DTS represents the Dominant Time Scale. DTS is the most probable time scale over which overflow occurs. In [SHRO03], a systematic way to derive DTS using real-time measurements is provided with the assumption that the input process to the multiplexing point in the network is Gaussian, which is by definition our assumption when employing (1), and we use this method for estimating DTS. With respect to w', there exist two options:

4.8.5.1.1.3.1 Option I

w' represents the mean inter-departure delay [TSE99], and it is defined as follows:

$$w' = \frac{h_{avg}}{N_{active}}, \qquad (7)$$

where N_{active} is the number of simultaneously active sources and h_{avg} is their average duration. If sources don't signal their termination, N_{active} could be provided as an estimate of the real number of simultaneously active sources using a time-out period, based on the activity of a source, as an indication of its termination.

In the case where we have various types of active sources with different average durations, we define w' as follows:

$$w' = \frac{h_{\min}}{N_{active}}, \qquad (8)$$

where $h_{\min}$ is the minimum average duration among all different types of active sources and N_{active} represents the total number of all active sources of all different types that are simultaneously active.

We select as measurement window the mean inter-departure delay (7), i.e. the time interval within which the system can be considered stationary -no flow departures-, or an even lower estimation of that interval (8) based on the worst case assumption – regarding flow departure dynamics - that all active sources have the same minimum average duration, unless this time interval is not long enough to capture the packet level dynamics of the aggregate traffic stream. In that case we use DTS as the value of the measurement window.

4.8.5.1.1.3.2 Option II

Assuming that by means of signalling the time instance when a source starts or stops transmitting is known, we can define w' as follows:

$$w' = \min\{t - t_{last\_adm,} t - t_{last\_end}\} \qquad (9)$$

where t corresponds to the time instance when an admission request arrives, t_{last_adm} corresponds to the time instance when the last call was admitted and t_{last_end} corresponds to the latest time instance when a call ceased. This expression for w' gives the exact time interval within which the system can be considered as stationary. But it can also be prone to significant measurement errors when w happens to be very small. As in option I, if sources don't signal their termination, a time-out period can be employed to indicate the termination of individual sources.

In this case we select as measurement window the exact time interval within which the system can be considered as stationary (9), unless this time interval is not long enough to capture the packet level dynamics of the aggregate traffic stream. In that case we use DTS as the value of the measurement window.

4.8.5.1.1.3.3 The Admission Control Criterion

Given the allocated bandwidth for the real-time traffic aggregate from edge-to-edge is C_{total}, and having computed the value for C_{est}, employing the precaution factor and the measurement window, the admission control criterion becomes:

$$\begin{array}{ll} If \quad (C_{est} \times PF) \le C_{total,} \quad admit \\ If \quad (C_{est} \times PF) > C_{total,} \quad reject \end{array} \qquad (10)$$

4.8.5.2 Elastic Traffic

The algorithm proposed for dealing with *SLS Invocation Handling* for real-time traffic cannot be applied to elastic traffic because the traffic pattern of TCP controlled traffic deviates greatly from the Gaussian assumption. For elastic traffic new algorithms must be derived taking into account previous work in this field, as in [FRE01], [PRAT00], [ROB98], [MAS99], [JOVE01], [CHAIT02], [CHARZ01].

4.8.6 Test Requirements

The SLS Invocation Handling algorithms must be tested with respect to the following:

- Support for different types of traffic
- Support for cSLSs/pSLSs
- Applicability for different traffic descriptors as these are depicted in the SLSs
- Interactions between different QCs
- Behaviour under different loading conditions.

The performance metrics against which the SLS Invocation Handling algorithms will be tested include the following:

- Loss, delay and jitter (real-time traffic)
- Throughput/Goodput (elastic traffic)
- QC utilisation (all types of traffic).

5 TRAFFIC ENGINEERING

5.1 Inter-domain TE terminology

5.1.1 Introduction

The purpose of this section is to clarify some of the technical terms involved in the description of the interactions between the various MESCAL functional blocks.

Figure 39 provides a graphical representation of the possible types of traffic, with respect to origin and destination, that may traverse an AS. It provides a basis for discussing and defining technical terms in the following Section.

The four types of traffic shown are as follows:

- (1) Traffic both originating and terminating in AS1;
- (2) Traffic originating in AS1 and terminating in a downstream AS;
- (3) Traffic originating in an upstream AS and terminating in a downstream AS;
- (4) Traffic originating in an upstream AS and terminating in AS1.

We define traffic that is originated and terminated within the same domain as *intra-domain traffic* (cases (1) and (4)), while traffic that terminates at a remote (downstream) domain is *inter-domain traffic* (cases (2) and (3)).

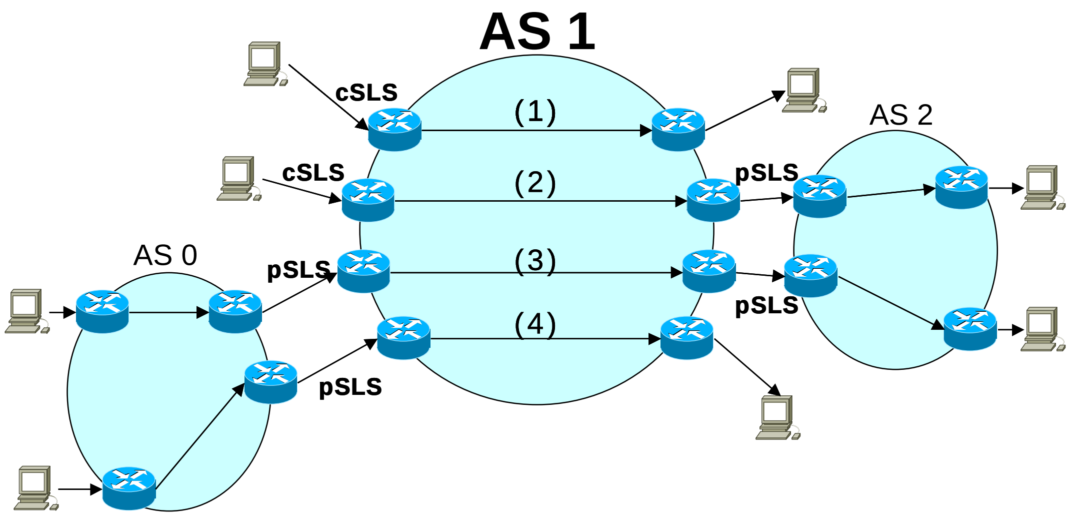


Figure 39. Traffic origination/destination cases

5.1.2 Definitions

5.1.2.1 pSLS$_{in}$ and pSLS$_{out}$

For a specific AS A:

- pSLS$_{in}$ denotes a pSLS that the domain A provides (or offers) to another AS B. That means that the domain A has received a request from domain B for the establishment of a pSLS.
- pSLS$_{out}$ denotes a pSLS that is provided to domain A by another domain B. That means that domain A has requested the establishment of a pSLS with domain B.

These terms are illustrated in Figure 40.

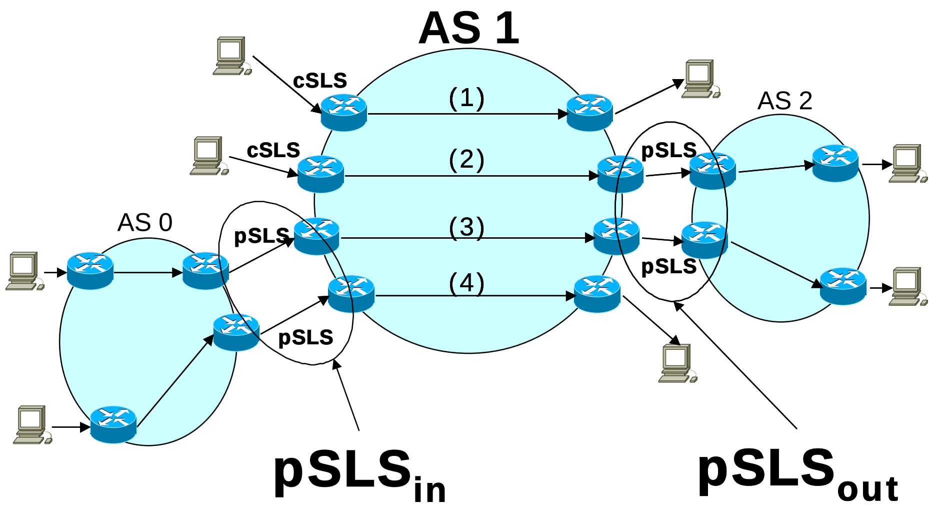


Figure 40. The $pSLS_{in}$ and $pSLS_{out}$ sets in a domain

5.1.2.2 eTM and iTM

For a specific AS A:

- eTM (external Traffic Matrix) denotes the traffic matrix that depicts the bandwidth requirements for all employed QCs, from each ingress interface (border router) to all the destination prefixes outside domain A. That is:

  eTM entry = [ingress i/f, {destination prefixes} Ï A, bw(min,max), e-QC]

  Considering Figure 39, the eTM represents the aggregate bandwidth requirements for traffic shown in types (2) and (3). Note that in the definition, {} represents the *set-of* semantics.

- iTM (internal Traffic Matrix) denotes the traffic matrix that depicts the bandwidth requirements for all employed QCs, from each ingress interface to all egress interfaces of domain A. The egress interfaces need not be the termination points of the traffic. That is:

  iTM entry = [ingress i/f, {egress i/f } Î A, bw(min,max), l-QC]

  Considering Figure 39, iTM represents the aggregate bandwidth requirements for traffic shown in all types (1)-(4).

5.1.2.3 eRAM, iRAM and RAM

For a specific AS A:

- eRAM (Resource Availability Matrix) represents the available resources, the bandwidth buffer (bw_buffer), for all QCs for all destination prefixes outside the domain A. eRAM must also specify the egress interface(s) (inter-domain links) that are used for providing these resources and additionally either the splitting ratio of the traffic on these egress interfaces or the mapping of the reachable destination prefixes on these egress interfaces. That is:

  eRAM entry = [ingress i/f, {egress i/f}+ratio, {destination prefixes} Ï A, bw_buffer, {l-QC}, e-QC]

- iRAM represents the available resources (bw_buffer) for all (local) QCs, from each ingress interface to all egress interfaces of domain A. The egress interfaces can be both termination points and transit points of traffic. That is:

  iRAM entry = [ingress i/f, {egress i/f}+ratio, bw_buffer, l-QC]

The term "bandwidth buffer" is defined below.

5.1.2.4 Bandwidth buffer

The bandwidth buffer, bw_buffer, is the bandwidth allocation computed by the traffic engineering processes. It is defined as the partitioning of the total available bandwidth between some *end points* for a given *QoS class*. The partitioning of the available bandwidth for a single QoS class reflects the provisioning decisions for that class. The bandwidth partitioning clearly defines the limits for admission control. The number of partitions depends on the policies used within the traffic engineering algorithms. In a simple scenario the total available bandwidth for a QoS class might be partitioned into two ranges, one that reflects the maximum bandwidth requirements of the currently subscribed SLSs, and the other that corresponds to either the provisioned bandwidth for future SLS subscriptions, or any overbooking ratios. For example a given bw_buffer=(10Mbps, 15Mbps) means that the total provisioned bandwidth for a particular QoS class is 15Mbps, from which 10Mbps corresponds to the requirements of currently subscribed flows.

The definition above assumes the bw_buffer is for a single *QoS class* between two *end points*. In the case where the bw_buffer is within an eRAM, the QoS class is the e-QC, and the end-points are the ingress interface and destination prefixes, of the corresponding eRAM entry. In the case where the bw_buffer is within an iRAM, the QoS class is the l-QC, while the end points are the ingress and the egress interfaces of the corresponding iRAM entry.

5.2 Traffic Forecast

5.2.1 Objectives

The main objectives of *Traffic Forecast* (TF) are:

- To forecast QoS traffic demand, based on existing and anticipated subscriptions, c/pSLS, related historical data combining subscriptions and network usage, and related business policies e.g. sale targets. This is required for the traffic engineering (TE) functions to appropriately dimension the network in terms of required intra- and inter-domain resources.
- To assess the validity of the forecasted traffic demand against actual usage statistics and based on various alarms coming from the service management and traffic engineering functions, and determine the cases where the current forecasts are no longer valid, triggering the process of revising them.
- To support the required interactions at RPC (resource provisioning cycles) epochs with the traffic engineering functions.

The main outcome of TF, of interest to MESCAL, is the production of the so-called:

- *Internal Traffic Matrix (iTM)*, presenting forecasted QoS traffic demand between network ingress and egress interfaces (border routers) of the domain, and the
- *External Traffic Matrix (eTM)*, presenting forecasted QoS traffic demand between network ingress interfaces and destination prefixes outside the domain.

Traffic demand is expressed in terms of bandwidth units. For scalability reasons, the QoS traffic needs to be of aggregate nature as well as the remote destinations should be coarsely defined. The iTM and eTM are specified from TE perspectives in Section 5.1

5.2.2 Interface Specification

The interfaces of *Traffic Forecast* with the rest of the components of the MESCAL functional architecture are shown in Figure 41 and briefly described in the following.

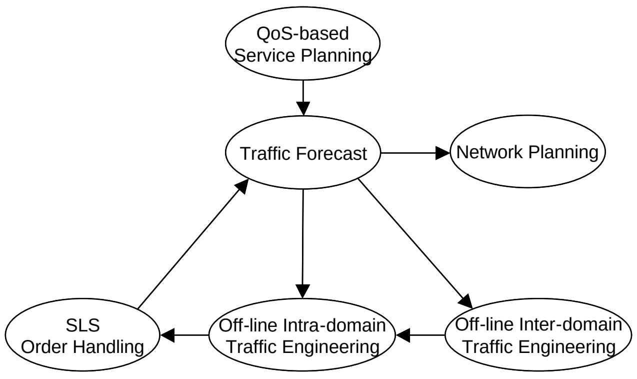

QoS-based Service Planning
Traffic Forecast
Network Planning
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering

Figure 41. Functional relationships of Traffic Forecast

Traffic Forecast receives from QoS-based Service Planning the expected demand for the QoS traffic as seen by business perspectives e.g. as a result of sale targets or newly launched marketing campaigns. In particular, the following information is passed:

- Traffic demand for currently supported and envisaged services between particular sets of source and destinations. Traffic demand in this case, may either be expressed directly, as a specific amount of bandwidth, or indirectly, as the number of new expected customers.
- Policy parameters related to the functionality of the Traffic Forecast process.

Traffic Forecast receives from SLS Order Handling information on negotiated services; specifically:

- Established service agreements, c/p SLS during the current and previous RPCs.
- Negotiation logs during the current RPC and previous RPCs.
- Various alarms on thresholds defined on the rate at which, service agreements are requested and established and related historical data.

Traffic Forecast provides to the Network Planning and Traffic Engineering components the traffic matrices it produces. This is to the end of ensuring that the local and inter-domain resources will be planned and subsequently engineered so that to effectively and gracefully accommodate established c/pSLSs as well as those anticipated to be ordered during the current provisioning cycle.

Network Planning aspects are outside the scope of MESCAL investigation. With respect to the two traffic matrices of interest to MESCAL outlined in the previous section, *Traffic Forecast* outputs the eTM to the *Off-line Inter-domain Traffic Engineering* component and the iTM to the *Off-line Intra-domain Traffic Engineering* component. While eTM calculation is solely based on the existing and anticipated population of subscriptions and related historical data, c/pSLS, the calculation of the iTM is additionally based on the outcome of the *Off-line Inter-domain Traffic Engineering* component. In fact, this interaction is of an iterative nature, internal to the traffic engineering algorithms (hence, not shown in the figure). All these interactions occur at RPC epochs.

Last, *Traffic Forecast* interacts with the network monitoring services to retrieve appropriate usage statistics to the end of validating its current forecast. This kind of interactions is not shown in Figure 41; network monitoring is outside the scope of MESCAL.

5.2.3 Behavioural Specification

5.2.3.1 Functional Decomposition

Figure 42 presents the internal functional architecture of the *Traffic Forecast* component, together with the interactions with the 'rest of the world' identified in the previous section.

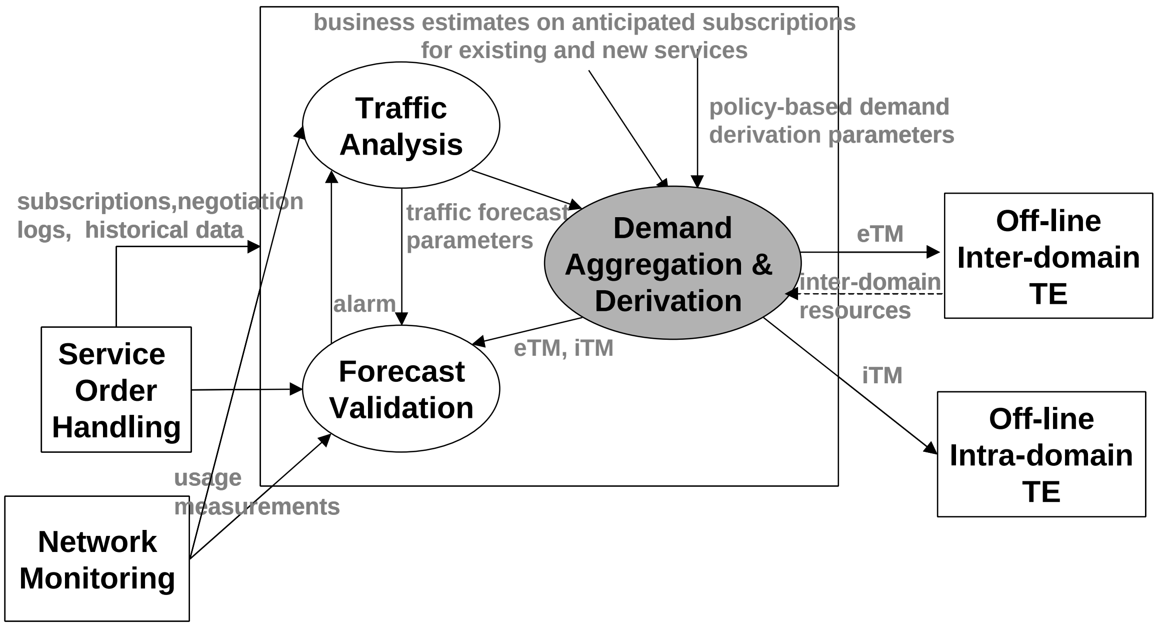

business estimates on anticipated subscriptions for existing and new services
policy-based demand derivation parameters
Traffic Analysis
subscriptions,negotiation logs, historical data
traffic forecast parameters
Demand Aggregation & Derivation
eTM
Off-line Inter-domain TE
inter-domain resources
alarm
eTM, iTM
iTM
Service Order Handling
Forecast Validation
Off-line Intra-domain TE
usage measurements
Network Monitoring

Figure 42. Functional decomposition of Traffic Forecast

The Traffic Analysis component is responsible for calculating the so-called *traffic forecast parameters*, which will allow the Demand Aggregation & Derivation component to calculate the required traffic matrices. This component is based on historical data regarding service requests/subscriptions and network usage. By employing appropriate statistical test and inference methods, together with ad-hop means, it determines the traffic forecast parameters, which correlate static aspects of the traffic implied by a population of service subscriptions, to traffic volumes offered to the network. The traffic forecast parameters include: the different service classes that could be distinguished from traffic forecast perspectives and associated multiplexing factors and aggregation weights; defined in section 5.2.3.2.1. These notions have been commonly used since the traditional telecom business world.

The Demand Aggregation & Derivation component is responsible for deriving the traffic demand implied by a specific population of service subscriptions, based on the forecast parameters produced by the Traffic Analysis component. First, the identified service classes are determined by the characteristics of the subscriptions; and subsequently, their anticipated demand is calculated per QoS class supported by the domain, based on the multiplexing factors and aggregation weights specified by the Traffic Analysis component. Note that this process is linear, thanks to the semantics of the forecast parameters.

The Forecast Validation component is responsible for assessing the validity of the traffic matrices produced by the Demand Aggregation & Derivation component. Validity is checked against actual traffic developments, by comparing the forecasted demand as specified in the traffic matrices with suitable extrapolations, in the medium to long term, derived by actual measurements of offered QoS traffic per service class. Furthermore, the component utilises feedback from the service order handling function regarding trends in the rate of service offer requests and established agreements. It also utilises feedback from the traffic engineering functions regarding significant under-utilisation or overloading of the resources of the dimensioned (against the traffic matrix under test) network.

MESCAL focuses only on the Demand Aggregation & Derivation component. The aspects involved in the other two components fall outside the scope of investigation of the project.

5.2.3.2 Demand Aggregation and Derivation

5.2.3.2.1 Notions and Terminology

The following traffic forecast parameters are considered:

- *Service Class (SC)*: The notion of service classes has been introduced to cope with the user diversity in service usage. Service classes distinguish offered services, p/cSLSs, based on their technical characteristics (e.g. invocation method, topological scope) according to the levels of statistical convergence observed in their usage patterns. As such, given a certain service class, it is considered that the users of the c/pSLSs of this class have the same service usage habits; hence valid multiplexing factors (see below) could be determined.
- *Multiplexing Factor (MF)*: For a given service class, a multiplexing factor is defined to be a proportion that consistently can relate the total subscribed traffic demand of a subscription population to the actual traffic peak that this population offers to the network. The validity of multiplexing factors should have been statistically verified over multiple observation periods involving subscription populations of different size. Given the variability in Internet service users, we take that a unique multiplexing factor per service class cannot be safely estimated; therefore, we assume two values for safely specifying a multiplexing factor: a minimum and a maximum, denoted by *Mfmin*, *Mfmax*, respectively.
- *Aggregation Weight (AW)*: For a given service class, the aggregation weight is the relative contribution of the service class's actual traffic peak to the peak of the corresponding offered total traffic. This notion is necessitated by the fact that the BHT (busy hour time, the period where traffic peak happened) of different service classes occurs in different time periods.

The following notions underline the functionality of the Demand Aggregation & Derivation component.

- *TT (traffic trunk)*: a traffic commodity that a provider domain must serve, therefore must be dimensioned against. It does not denote a commodity offered by the network; in fact, TTs multiplex the offered commodities c/SLS. With respect to the MESCAL QoS terminology, a TT is defined as a QoS-class in a certain topological scope. We naturally then, distinguish between *local TTs (l-TTs)* and external *TTs (e-TTs)* as follows:

- *l-TT = <ingress i/f, egress i/f, l-QC>*, where the ingress and egress i/f exist in the boundaries of the domain
- *e-TT = <ingress i/f, ['dest'], e-QC>*, where 'dest' denotes a destination prefix outside the domain.

Based on the above definitions, eTM refers to e-TTs, while iTM refers to l-TTs

- *Virtual-Egress (VE)*: is a virtual node in the domain, which corresponds to a pSLS the provider is bound to i.e. is an ‘extra’ leaf-node connected to the border g/w node that interconnects the provider with the provider the pSLS is established with. The corresponding link is called a *Virtual-Link (VE)*.

5.2.3.2.2 eTM, iTM Calculation

This section outlines the process of calculating eTM and iTM. This process executes at the beginning of an RPC.

Consider a population of subscriptions, c/pSLS; this population may correspond to the union of already established and anticipated e.g. by market estimates c/pSLS. We assume that the c/pSLS have been mapped to SSS/SLS (by the Service Order handling component).

With reference to the terminology introduced in the previous section, it is worth-noting that:

- A c/pSLS corresponds uniquely to a service class, therefore its MF and AW can be determined.
- A c/pSLS can generate traffic for a number of TTs. If a c/pSLS entails only one QoS level it will generate traffic for only one l-TT if it is an intra-domain service, whilst for one e-TT if it is an inter-domain service. If the c/pSLS entails multiple QoS levels the number of l-TTs or e-TTs will correspondingly increase. Therefore, a c/pSLS may contribute to a number of entries in the iTM and/or the eTM.

/* 1 * Aggregation; c/pSLS are aggregated to TTs */

```
For each c/pSLS (SSS)
    For each SLS
        Determine service class and associated traffic forecast parameters (MF, AW)
        Determine TTs
            If c/pSLS is inter-domain specify e-TT (ingress i/f, [dest], e-QoS)
            If c/pSLS is intra-domain specify l-TT (ingress i/f, egress i/f, l-QC)
        Find Maximum Contracted Demand per TT (assuming a fluid-flow model)
        Find Anticipated Demand per TT (divide by MF and multiply by AW)
```

/* 2 * eTM */

From all determined e-TTs determine new set of e-TTs by aggregating their [dest] on the basis of LPM (longest prefix matching), for the same e-QC and calculate the Anticipated Demand of these new e-TTs by summing the demands of their member TTs.

Provide eTM to the *Off-line Inter-domain Traffic Engineering* component

Note: It is recommended that offered p/cSLSs do not have overlapping destination prefixes.

/* 3 * intermediate iTM */

For each l-TT (all those determined by step 1)

    Calculate Total Anticipated Demand per l-TT (sum-up Anticipated Demand per l-TT per SLS)

For each hose l-TT

    Calculate hoses and Total Anticipated Demand per hose   (topological aggregation of hoses)

Store the intermediate iTM.

/* 4 * complete iTM; it is assumed that the Off-line Inter-domain Traffic Engineering component has determined the pSLS$_{out}$ for accommodating eTM */

For each e-TT (all those determined by step 2)

    Based on inter-domain resource allocation i.e. determined pSLS$_{out}$ and the concept of Virtual-Edges, determine corresponding l-TT and the imposed demand

    Calculate Total Anticipated Demand per l-TT (sum-up Anticipated Demand per l-TT as calculated previously with the corresponding demand of the intermediate iTM cf. step 3 and update iTM)

Provide iTM to the *Off-line Intra-domain Traffic Engineering* component

5.2.4 Test Requirements

The previously specified process will be implemented. Tests will be carried out in a simulated environment with the purpose to:

- Prove-of-concept; can the identified notions and abstraction lead to a feasible design and implementation?
- Functional validity; does the process produce the expected results?
- Scalability assessment; processing time, memory requirements as a function of the environmental variables such as number of service classes, number of TTs, population of p/cSLS.

5.3 Traffic Engineering Interactions

5.3.1 Decomposition of Offline Inter-domain Traffic Engineering

Figure 43 shows the decomposition of the Offline Inter-domain Traffic Engineering component of the MESCAL functional architecture.

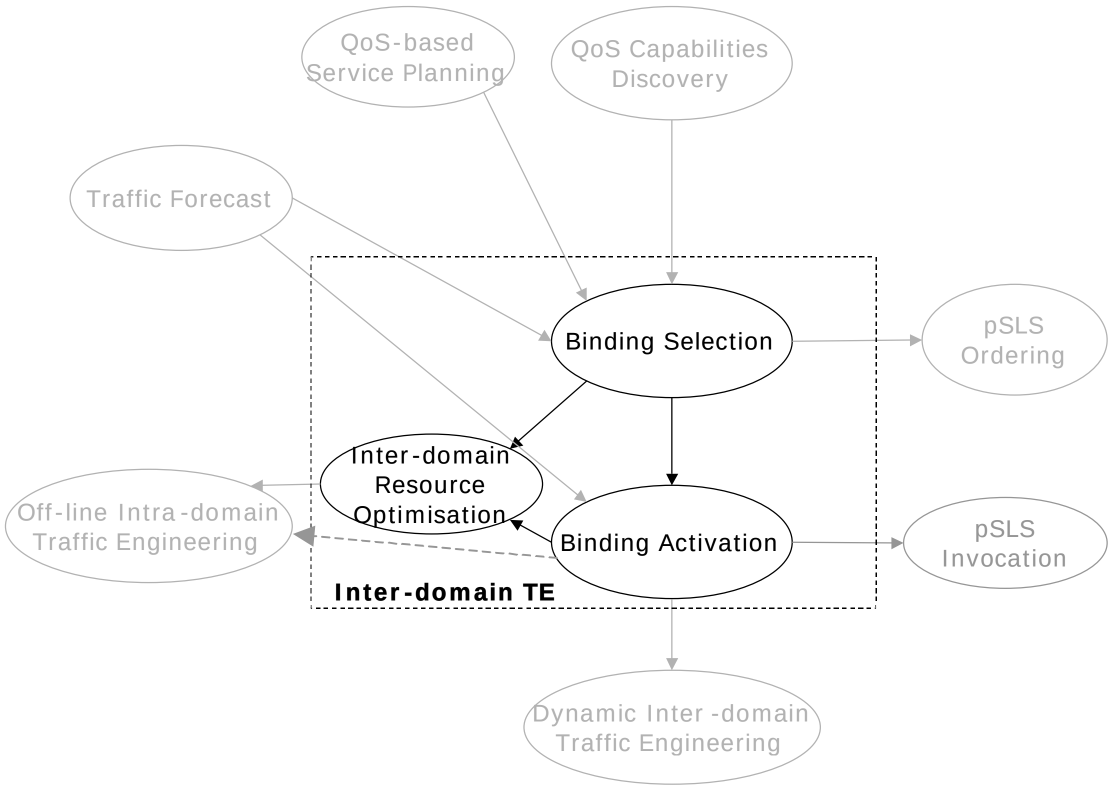

QoS-based Service Planning
QoS Capabilities Discovery
Traffic Forecast
Binding Selection
pSLS Ordering
Inter-domain Resource Optimisation
Off-line Intra-domain Traffic Engineering
Binding Activation
pSLS Invocation
Inter-domain TE
Dynamic Inter-domain Traffic Engineering

Figure 43. Decomposition of Offline Inter-domain TE

5.3.2 Resource Provisioning Cycles

5.3.2.1 Definitions

We define two terms as follows: the Intra-domain resource provisioning cycle (RPC) is the sequence of network resource dimensioning functions performed *within* a domain, while Inter-domain RPC is the sequence of network resource dimensioning functions performed *between* adjacent domains. Both Intra-domain RPC and Inter-domain RPC functions are performed at regular periods.

In the case of Inter-domain RPC, we need to further define two different cycles: the Binding Selection Cycle and the Binding Activation Cycle (Figure 44[1]).

The Binding Selection Cycle concerns the period when the Binding Selection component decides inter-domain resource usage, and determines which pSLSs to establish with the domain's peer ASs; the Binding Selection function block then commands the pSLS Ordering function block to negotiate these pSLSs with peer domains.

The Binding Activation Cycle is the period between two successive network resource dimensioning enforcements between adjacent domains, when this resource dimensioning is constrained by established pSLSs.

[1] Note that the arrows from Binding Selection and Binding Activation to SLS Order Handling do not appear in the overall functional architecture figure. They are illustrated here because the eRAM/iRAM are assumed to be sent via the Resource Optimisation and Intra-domain offline TE function blocks

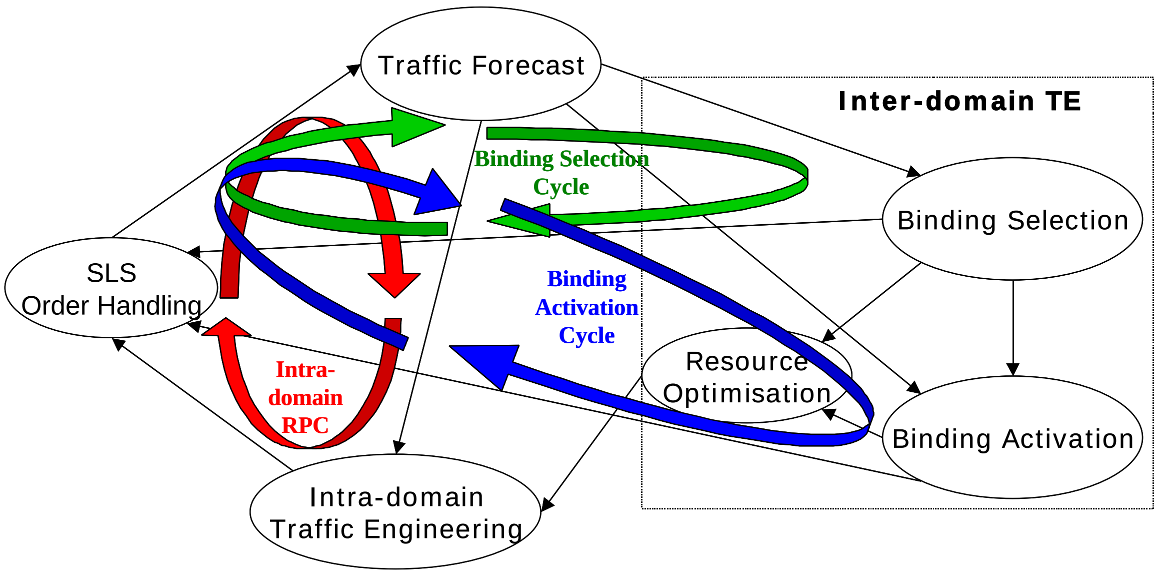


Figure 44. Resource Provisioning Cycles

5.3.2.2 Issues

The Binding Selection Cycle occurs less frequently and with a longer timescale than the Binding Activation Cycle; for example the Binding Selection Cycle may run monthly while the Binding Activation Cycle may run daily. Although it would provide a more optimal solution with regards to resource utilisation when the periods of both the Binding Selection Cycle and the Binding Activation Cycle are equal, it is not expected that ASs will change their established pSLSs with their peers every time they enforce a new intra/inter-domain resource configuration. In consequence, the MESCAL approach allows domains flexibility by assuming that the Binding Activation Cycle occurs more frequently than the Binding Selection Cycle. Every time a Binding Selection Cycle runs, a Binding Activation Cycle is also triggered so that the inter-domain configuration selected by Binding Activation reflects the current set of pSLSs.

We also argue that the periods of intra-domain and inter-domain RPC should be equal. The justification of this argument is given by analysing the consequences of the following possible combinations:

- Case 1: Intra-domain RPC occurs more often than Binding Activation Cycle

  In this case, intra-domain traffic changes more dynamically than inter-domain traffic. However, triggering intra-domain network dimensioning more frequently and independently from inter-domain dimensioning is not an optimal solution. This is because, since the inter-domain dimensioning remains unchanged, the resources allocated within the domain for inter-domain traffic remain intact. This gives more stringent link capacity constraints for Intra-domain network dimensioning subsystem to engineer the network for intra-domain traffic. With these constraints, the Intra-domain network dimensioning subsystem may not produce a network configuration that achieves optimal resource utilisation. As a result, only a sub-optimal intra-domain configuration can be achieved.

- Case 2: Binding Activation Cycle occurs more often than Intra-domain RPC

  This case is similar to case 1. Inter-domain traffic engineering may not be able to select an optimal egress point for inter-domain traffic due to the stringent link capacity constraints. In this case, the stringent capacity constraints are a result of fixed link resource allocations for the intra-domain traffic. As a result, only a sub-optimal inter-domain configuration can be achieved.

- Case 3: Equal Binding Activation Cycle and Intra-domain RPC periods

  When both the Binding Activation Cycle and Intra-domain RPC periods are equal, a network configuration that yields optimal resource utilisation for intra-domain and inter-domain traffic simultaneously is possible, by taking the latest iTM and eTM into consideration. In this case, compared to case 1 and 2, a more optimal intra-domain and inter-domain resource utilisation can be achieved.

5.3.3 Decoupled and integrated approaches to Inter- and Intra-domain TE

The purpose of this section is to introduce two potential approaches for inter-domain resource optimisation, namely decoupled and integrated resource optimisation. In decoupled resource optimisation, the algorithms that perform Inter-domain resource allocation and Intra-domain resource allocation run independently. The algorithm proceeds by iterating between inter- and intra-domain resource allocation. In comparison, the integrated resource optimisation approach considers both inter- and intra-domain resources at the same time.

In principle, the integrated approach will provide a more optimal system configuration since it is taking account of many variables simultaneously. However, the decoupled approach allows algorithms for inter- and intra-domain traffic engineering to be considered separately, and it is the view of the MESCAL team that this approach will initially lead to more fruitful insights into inter-domain QoS engineering. In this Section, we compare the two approaches. However, the algorithms that have been developed and which are described in Section 5.4.3 assume the decoupled approach. For each approach, we provide a brief introduction, and describe the inputs and outputs, and give a process description.

5.3.3.1 Decoupled Inter-domain Resource Optimisation

Figure 45 shows the generic decoupled *Inter-domain Resource Optimisation* approach. The description of this approach is as follows.

Two function blocks are shown in the figure: *Inter-domain Resource Optimisation* and *Offline Intra-domain TE*. The function of *Offline intra-domain TE* is to compute the intra-domain network configuration and dimension resources between edges within a network.

Initially, *Inter-domain Resource Optimisation* accepts some input data and actions from *Binding Activation* or *Binding Selection*. It is noted that *Inter-domain Resource Optimisation* is responsible for mapping customer inter-domain traffic to appropriate egress points/pSLSs while satisfying the traffic with QoS requirements. In order to satisfy the traffic with QoS requirements, both inter-domain and intra-domain must have sufficient resources to accommodate the traffic. Since the availability of inter-domain resource (i.e. pSLSs$_{out}$) has been produced by *Binding Selection* and is passed to *Inter-domain Resource Optimisation* through *Binding Activation*, *Inter-domain Resource Optimisation* can simply check whether or not a specific pSLS$_{out}$ has sufficient resource to accommodate the traffic with QoS requirements. However, *Inter-domain Resource Optimisation* is not able to determine whether resources within the domain are sufficient or not. Since *Offline Intra-domain TE* is responsible for routing traffic within the network towards specified destination or egress points, it is able to return the resulting intra-domain resource availability to *Inter-domain Resource Optimisation,* which in turn can select an appropriate inter-domain TE solution. Thus, communication between *Inter-domain Resource Optimisation* and *Offline Intra-domain TE* is necessary.

Basically, there are two reasons to establish communication between *Inter-domain Resource Optimisation* and *Offline Intra-domain TE*. The first one is that, as discussed in the previous paragraph, *Offline Intra-domain TE* can help to check whether there is sufficient intra-domain resource to accommodate the traffic with QoS requirements. The second reason is that *Inter-domain Resource Optimisation* can assign egress points to inter-domain traffic as a presumed solution and request *Off-line intra-domain TE* to indicate the resulting intra-domain resource availability and utilisation after both intra-domain and inter-domain traffic (egress points have been identified from that presumed solution) are being routed in the network. Specifically, this evaluates the impact of that presumed solution when routed with intra-domain traffic on the intra-domain resource utilisation. This reason is supported when the objective of *Inter-domain Resource Optimisation* is to optimise not only inter-domain resource utilisation but also intra-domain resource utilisation.

With the above reasoning, *Inter-domain Resource Optimisation* queries *Offline Intra-domain TE* to perform intra-domain resource checking and to provide a resulting intra-domain resource utilisation by giving a presumed inter-domain TE solution (eRAM). The resulting intra-domain resource availability and utilisation returned from off-line intra-domain TE will help *Inter-domain Resource Optimisation* to determine an optimal inter-domain TE solution while meeting the target optimisation objectives. *Inter-domain Resource Optimisation* may consult with *Offline Intra-domain TE* a number of times in order to produce a single or multiple optimal inter-domain TE solutions.

From *Inter-domain Resource Optimisation* point of view, *Offline Intra-domain TE* is a black box which only provides interfaces to accept input, make decisions and produce output. The *Offline Intra-domain TE* can be any existing intra-domain TE solutions such as OSPF together with Constrained Shortest Path First (CSPF) or the TEQUILA intra-domain TE subsystem [TEQUILA]. In this decoupled approach, *Inter-domain Resource Optimisation* and *Offline Intra-domain TE* operate separately but a relationship by functional calls and parameters passing is established between them.

The next section describes input, output and processes of the generic decoupled *Inter-domain Resource Optimisation* approach.

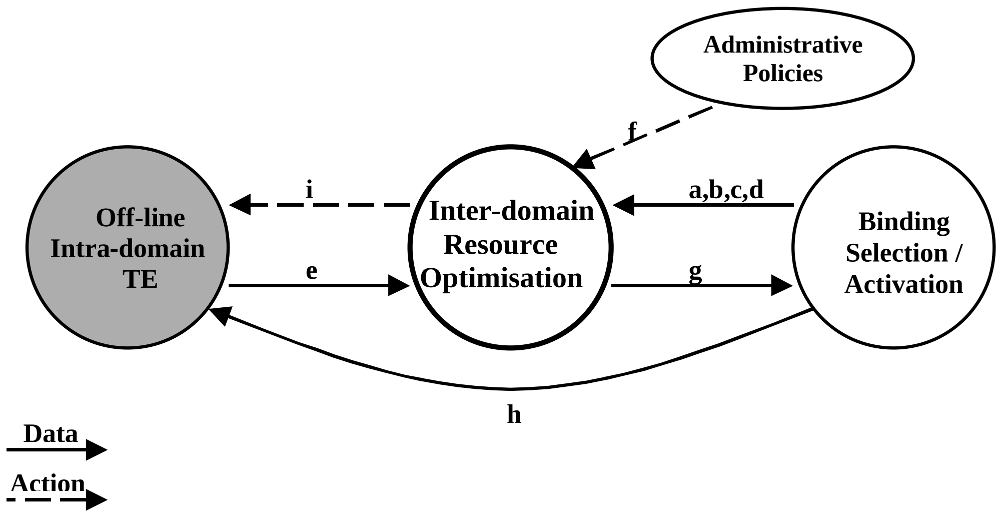


Figure 45. Decoupled Inter-domain Resource Optimisation

5.3.3.1.1 Decoupled Inter-domain Resource Optimisation input and output

This section describes input and output of the decoupled *Inter-domain Resource Optimisation*. Input and output are further divided into two types: data and action. Action requires some controls or feedback. The letter used in each item below corresponds to the one in Figure 45 and the target of usage (either *Binding Selection* or *Binding Activation*) is specified in a square bracket. If the target of usage is not specified, the corresponding item is applicable to both the *Binding Selection* and *Binding Activation*.

<u>Input data</u>

a) [*Binding Selection*] *eTM* -- The extended traffic matrix produced by *Traffic Forecast* and it is passed to *Inter-domain Resource Optimisation* through *Binding Selection*. The time scale of this eTM is large.

[*Binding Activation*] *eTM* -- The extended traffic matrix produced by *Traffic Forecast* and it is passed to *Inter-domain Resource Optimisation* through *Binding Activation*. The time scale of this eTM is small.

b) [*Binding Activation*] *pSLSs$_{out}$* -- A set of established pSLSs$_{out}$ produced by *Binding Selection*. The required fields include destination prefix, o-QC, bandwidth availability and egress interface.

c) [*Binding Selection*] One or more sets of options for sets of l-QCs, o-QCs and egress node IDs.

[*Binding Activation*] *QC mapping compatibility* -- A corresponding QC mapping compatibility for each pSLS$_{out}$. The mapping compatibility describes a set of eligible l-QCs maps to a specific o-QC.

d) Number of solutions to be returned.

e) *iRAM:* The internal resource availability matrix that specifies estimates of the availability of the engineered network to accommodate QoS traffic between edges in the network.

<u>Input action</u>

f) *Policies*: customised policies that may affect inter-domain resource optimisation decisions.

<u>Output data</u>

g) [*Binding Selection*] A number of *Network configuration* according to the parameter of *number of solutions to be returned*. Each network configuration is associated with a cost value to access its quality.

[*Binding Activation*] *eRAM(s)* -- The extended resource availability matrix produced by *Inter-domain Resource Optimisation*. It is an inter-domain traffic engineering solution that specifies estimates of the availability of the inter-domain resources (e.g. pSLSs$_{out}$) to accommodate QoS traffic towards the upstream service-peering domains. Each solution is associated with a cost value to access its quality. *Inter-domain Resource Optimisation* produces a set of eRAM(s) (i.e. inter-domain TE solutions) corresponds to the same input data according to the parameter of *number of solutions to be returned*.

h) [*Binding Activation*] *eRAM*: The best inter-domain TE solution selected by *Binding Activation* (eRAM$\in$eRAM(s)). Off-line intra-domain TE has to be informed this selected solution in order to select the corresponding intra-domain configuration.

<u>Output action</u>

i) *Intra-domain resource query*: A "what-if"-type query which includes a solution of *Binding Selection* or *Binding Activation* as a parameter and asks for internal network resource availability if the input solution is presumably selected/activated.

5.3.3.1.2 Decoupled Inter-domain Resource Optimisation process

This section briefly describes how the decoupled *Inter-domain Resource Optimisation* works in general. The process of decoupled *Inter-domain Resource Optimisation* is divided into following steps (words in bold are data or actions that have been defined in the previous section):

Calls from Binding Selection

1. Inter-domain *Resource Optimisation* receives **eTM, One or more sets of options for sets of l-QCs, o-QCs and egress node IDs** and the **number of solutions to be returned** from *Binding Selection* as input data and **policies** (if any) as an input action.

2. The decision of *Inter-domain Resource Optimisation* may depend on intra-domain resource availability and utilisation. *Inter-domain Resource Optimisation* sends an **intra-domain resource query** to *Offline Intra-domain TE* requesting the resulting intra-domain resource availability and utilisation assuming a given *Binding Selection* solution were to be selected.

3. *Offline Intra-domain TE* takes the solution as input and then uses its optimisation algorithms to compute a network configuration for both intra-domain and inter-domain traffic routed within

the network. *Offline Intra-domain TE* answers the query by providing **iRAM** as output to *Inter-domain Resource Optimisation*.

4. *Inter-domain Resource Optimisation* takes iRAM into consideration (together with other factors) to determine whether the considered solution is an optimal solution while achieving the target resource utilisation objectives.
5. *Inter-domain Resource Optimisation* may produce multiple solutions by repeating the step from 2 to 4 with different *Binding Selection* solutions as input.
6. *Inter-domain Resource Optimisation* outputs a set of solutions to *Binding Selection* for which to select and implement the best one.

Calls from Binding Activation

1. *Inter-domain Resource Optimisation* receives eTM, a set of pSLSsout , the corresponding QC mapping compatibility and the number of solutions to be returned from *Binding Activation* as input data and policies (if any) as an input action.
2. The decision of *Inter-domain Resource Optimisation* may depend on intra-domain resource availability and utilisation. *Inter-domain Resource Optimisation* sends an intra-domain resource query to *Offline Intra-domain TE* requesting the resulting intra-domain resource availability and utilisation assuming a given inter-domain TE solution (eRAM) were to be put into effect or activated.
3. *Offline Intra-domain TE* takes this given eRAM as input and then uses its optimisation algorithms to compute a network configuration for both intra-domain and inter-domain traffic routed within the network. *Offline Intra-domain TE* answers the query by providing iRAM as output to the *Inter-domain Resource Optimisation*.
4. *Inter-domain Resource Optimisation* takes iRAM into consideration (together with other factors) to determine whether the considered inter-domain TE solution is an optimal solution while achieving the target resource utilisation objectives.
5. *Inter-domain Resource Optimisation* may produce multiple optimal inter-domain TE solutions by repeating the step from 2 to 4 with different eRAM as input.
6. *Inter-domain Resource Optimisation* outputs a set of solutions (eRAM(s)) to *Binding Activation* for which to select and implement the best one.
7. Finally, *Binding Activation* informs *Offline Intra-domain TE* of the selected inter-domain TE solution ($eRAM \in eRAM(s)$) for which to select the intra-domain configuration corresponds to that selected solution.

5.3.3.2 Integrated Inter-domain Resource Optimisation

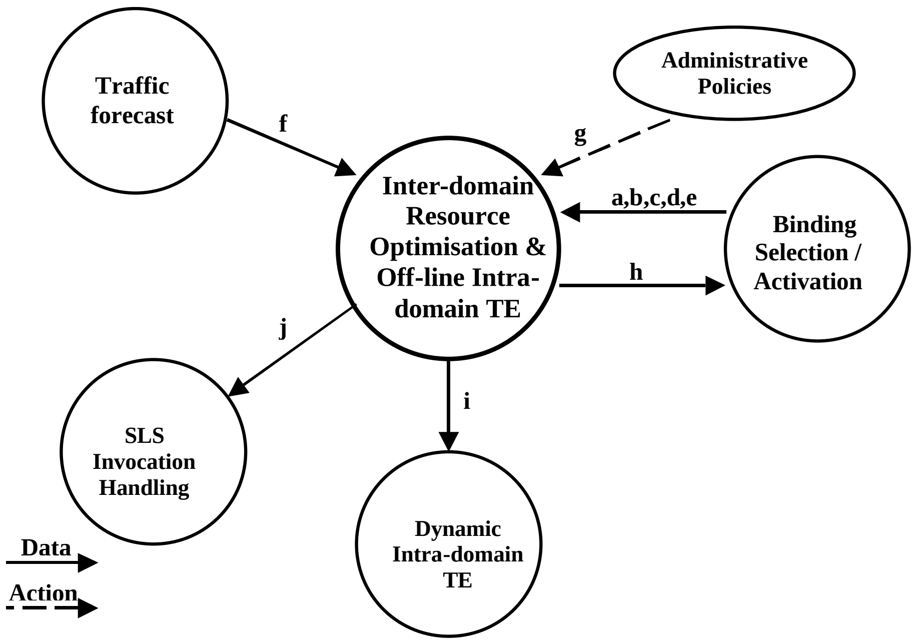


Figure 46. Integrated Inter-domain Resource Optimisation

Figure 46 shows the generic integrated *Inter-domain Resource Optimisation* approach. This approach integrates *Inter-domain Resource Optimisation* with *Offline Intra-domain TE* to produce a complete traffic engineering solution for both intra-domain and inter-domain traffic simultaneously. The integrated approach differs from the decoupled approach in that the integrated approach, unlike the decoupled approach, does not presumably assign egress points to inter-domain traffic and consult with *Offline Intra-domain TE* to obtain the resulting intra-domain resource utilisation. Instead, the integrated approach performs inter-domain TE *collaboratively* with intra-domain TE to produce a complete traffic engineering solution simultaneously (this includes egress point/pSLS$_{out}$ selection and traffic routing between edge nodes in the network).

Note that, in the integrated approach, since inter-domain traffic is routed in the network where some resources are consumed, each decision on routing inter-domain traffic may affect the decision of routing intra-domain traffic in the network since both types of traffic share resources within the network and the capacity of those resources are constrained. This also holds for the opposite case where each decision on routing intra-domain traffic can affect the decision of egress point selection for inter-domain traffic.

5.3.3.2.1 Integrated Inter-domain Resource Optimisation input and output

This section describes input and output of the integrated *Inter-domain Resource Optimisation*. Input and output are further divided into two types: data and action. Action requires some controls or feedback. The letter used in each item below corresponds to the one in Figure 46 and the target of usage (either *Binding Selection* or *Binding Activation*) is specified in a square bracket. If the target of usage is not specified, the corresponding item is applicable to both *Binding Selection* and *Binding Activation*.

<u>Input data</u>

a) [*Binding Selection*] *eTM* -- The extended traffic matrix produced by *Traffic Forecast* and it is passed to *Inter-domain Resource Optimisation* through *Binding Selection*. The time scale of this eTM is large.

   [*Binding Activation*] *eTM* -- The extended traffic matrix produced by *Traffic Forecast* and it is passed to *Inter-domain Resource Optimisation* through *Binding Activation*. The time scale of this eTM is small.

b) [*Binding Activation*] *pSLSs$_{out}$* -- A set of established pSLSs$_{out}$ produced by *Binding Selection*. The required fields include destination prefixes, o-QC, bandwidth availability and egress interface.

c) [*Binding Selection*] One or more sets of options for sets of l-QCs, o-QCs and egress node IDs.

[*Binding Activation*] *QC mapping compatibility* -- A corresponding QC mapping compatibility for each pSLS$_{out}$. The mapping compatibility describes a set of eligible l-QCs maps to a specific o-QC.

d) Number of solutions to be returned.

e) [*Binding Activation*] *eRAM*: The selected inter-domain TE solution, eRAM$\in$eRAM(s), which is determined by *Binding Activation.*

f) *iTM:* The internal traffic matrix that includes traffic routed between edges in a network. Note that this iTM does not contain any inter-domain traffic since its egress point has not been selected yet.

<u>Input action</u>

g) *Policies*: customised policies that may affect *Inter-domain Resource Optimisation* decisions.

<u>Output data</u>

h) [*Binding Selection*] A number of *Network configuration* according to the parameter of *number of solutions to be returned*. Each network configuration is associated with a cost value to access its quality.

[*Binding Activation*] *eRAM(s)* -- The extended resource availability matrix produced by *Inter-domain Resource Optimisation*. It is an inter-domain traffic engineering solution that specifies estimates of the availability of the inter-domain resources (e.g. pSLSs$_{out}$) to accommodate QoS traffic towards the upstream service-peering domains. Each solution is associated with a cost value to access its quality. *Inter-domain Resource Optimisation* produces a set of eRAM(s) (i.e. inter-domain TE solutions) corresponds to the same input data according to the parameter of *number of solutions to be returned*.

i) [*Binding Activation*] *iRAM:* The internal resource availability matrix that specifies estimates of the availability of the engineered network to accommodate QoS traffic between edges in the network.

j) [*Binding Activation*] *RAM:* RAM should give an estimate of the available resources for the various reachable destination prefixes (end-to-end) for all employed QCs or Meta-QoS-Classes. The estimates could be expressed as a range of bandwidth values, taking into account potential resource sharing between classes and intra/inter-domain reachable destination prefixes.

5.3.3.2.2 Integrated Inter-domain Resource Optimisation process

This section briefly describes how the integrated *Inter-domain Resource Optimisation* works in general. The process is divided into following steps: (words in bold are data or action that have been defined in the previous section)

Calls from Binding Selection

1. The integrated *Inter-domain Resource Optimisation* receives **eTM**, **One or more sets of options for sets of l-QCs, o-QCs and egress node IDs** and **iTM** as input data from *Binding Selection* and *Traffic Forecast*, and **policies** (if any) as input action.
2. The integrated *Inter-domain Resource Optimisation* takes the input and computes a complete traffic engineering solution simultaneously for both intra-domain and inter-domain traffic. This includes egress point/pSLS$_{out}$ selection, traffic routing between edge nodes in the network. The integrated inter-domain resource optimisation may produce a set of optimal inter-domain and intra-domain traffic engineering solutions.

3. The integrated *Inter-domain Resource Optimisation* outputs a set of solutions to *Binding Selection* for which to select and implement the best one.

Calls from Binding Activation

1. The integrated *Inter-domain Resource Optimisation* receives **eTM**, a set of **pSLSs$_{out}$**, the corresponding **QC mapping compatibility** and **iTM** as input data from *Binding Activation* and *Traffic Forecast*, and **policies** (if any) as input action.
2. The integrated *Inter-domain Resource Optimisation* takes the input and computes a complete traffic engineering solution simultaneously for both intra-domain and inter-domain traffic. This includes egress point/pSLS$_{out}$ selection, traffic routing between edge nodes in the network. The integrated *Inter-domain Resource Optimisation* may produce a set of optimal inter-domain and intra-domain traffic engineering solutions. Each solution consists of eRAM with corresponding iRAM (or collectively called RAM).
3. The integrated *Inter-domain Resource Optimisation* outputs a set of optimal inter-domain TE solutions (**eRAM(s))** to *Binding Activation* for which to select and implement the best solution. *Binding Activation* returns the selected inter-domain TE solution, **eRAM**$\in$eRAM(s), to the integrated *Inter-domain Resource Optimisation*.
4. According to the selected eRAM, the integrated *Inter-domain Resource Optimisation* selects the corresponding **iRAM**. The iRAM is output to dynamic intra-domain TE for which to configure the network while **RAM** (the selected eRAM plus its corresponding iRAM) is output to *SLS Invocation Handling* for which to control the amount of traffic injected into the network.

5.3.4 Off-line inter-domain TE cases

This section explores all the possible cases for off-line inter-domain traffic engineering. Readers are referred to D1.1 section 5.4 for relevant detail [D1.1]. The following cases are applied to each inter-domain traffic (i.e. aggregated traffic based on ingress router and destination prefix).

5.3.4.1 Single/Multiple egress point selection

Under inter-domain routing, it is common that a destination prefix can be reached through multiple egress points in a network. Service and Network Providers thus have to select appropriate egress points to egress inter-domain traffic. Two variant of egress point selection are deduced. For ***single egress point selection***, only a single egress point is selected for each destination prefix. Thus, all traffic towards a destination prefix, regardless of using which ingress routers, will always egress from the same egress point. In practice, single egress point selection is used when Service and Network Providers always have a preferable egress point over the others for each destination prefix. However, it is possible to improve network resource utilisation by allowing multiple egress points for each destination prefix. In this case, ***multiple egress point selection***, all traffic towards a destination through a designated ingress router will egress from a selected egress point. As a result, multiple egress point selection allows resource load balancing and the assignment of an optimal egress point to each aggregated inter-domain traffic based on ingress router and destination prefix.

5.3.4.2 Single/Multiple pSLS$_{out}$ selection

Although an egress point is selected, there may still be multiple pSLSs$_{out}$ that are offered by the same or different service peering providers and are attached with the egress point towards a destination prefix. Service and Network Providers thus have to select appropriate pSLSs$_{out}$ to egress traffic to the destination prefix with end-to-end guarantees. Two variant of pSLS$_{out}$ selection are deduced. For ***single pSLS$_{out}$ selection***, only a single pSLS$_{out}$, among all the pSLSs$_{out}$ that are attached with each egress point, is selected. In this case, load balancing between multiple pSLSs$_{out}$ on each egress point is not allowed. However, single pSLS$_{out}$ selection exists for policy and managerial reasons and when the cost to sign a pSLS$_{out}$ is high. On the other hand, ***multiple pSLS selection*** allows more than one

$pSLS_{out}$ to be selected among all the $pSLSs_{out}$ that are attached with each egress point. In this case, load balancing between multiple $pSLSs_{out}$ on each egress point is allowed.

5.3.4.3 Single/Multiple l-QC selection

To provide an extended QC, a Service or Network Provider has to bind its l-QC with the QC offered by its service peering provider. There may be multiple appropriate l-QCs to complete the binding. ***Single l-QC selection*** allows only a single l-QC bind to each offered QC, while ***multiple l-QC selection*** allows more than one l-QC for the reason of intra-domain load balancing.

5.3.4.4 Single/Multiple intra-domain route selection

When an egress point has been chosen, a path is then found between the designated ingress router and the chosen egress point. By taking a selected l-QC into consideration, there may be multiple paths to the egress point. ***Single intra-domain route selection*** allows only a single path selected between the ingress router and the egress point, while ***multiple intra-domain route selection*** allows multiple paths for the reason of intra-domain load balancing.

5.3.4.5 QoS parameters consideration

We consider a set of binding activation problems by considering various QoS parameters. Three possible binding activation problems can be deduced:

- ***Bandwidth constrained binding activation***: Map the predicted traffic matrix to the inter-domain network resources, satisfying bandwidth requirements while aiming at optimising the use of network resources.
- ***Delay constrained binding activation***: Map the predicted traffic matrix to the inter-domain network resources, satisfying delay requirements while aiming at optimising the use of network resources.
- ***Jitter/Loss constrained binding activation***: Map the predicted traffic matrix to the inter-domain network resources, satisfying jitter/loss requirements while aiming at optimising the use of network resources.

5.3.4.6 Traffic engineering scenarios

All the potential problems defined in the Sections 5.3.4.1 to 5.3.4.4 can be combined to form a set of traffic engineering scenarios. The justification is that:

- The selection of egress point does not mean that pSLS will also be chosen and vice versa. The reader is referred to D1.1 section 5.4.2.2 for more detail [D1.1].
- The selection of both egress points and $pSLSs_{out}$ means that an exit point to egress traffic towards a destination prefix has been identified. The next step is to bind l-QCs to these egress points and pSLSs.
- The selection of l-QCs may be supported by multiple intra-domain paths between the ingress and egress point.

Thus, there are maximum 2^4 possible traffic engineering scenarios formed by the combination of problems defined in Sections 5.3.4.1 to 5.3.4.4. Note that, however, some of the scenarios may not be available for each of the three MESCAL solution options. The number of scenarios becomes larger if we consider more QoS parameters. We formulated three binding activation problems in section 3.5 and these problems, when arbitrary combined together, form a multi-constrained problem. Thus, this produces a set of single/multi-constrained problems by combination. We denote N by the number of problem combinations in section 3.5. In this case, the maximum number of possible traffic engineering scenarios becomes $N \cdot 2^4$.

5.4 Offline Inter-domain TE

5.4.1 Binding Selection

5.4.1.1 Introduction

The *Binding Selection* function block is part of the Offline Inter-domain Traffic Engineering function. The component that implements these functions will run at the Binding Selection Cycle epoch. As described in Section 5.3.2, *Binding Selection* is expected to run less frequently than *Binding Activation*.

The functions of *Binding Selection* may be divided into three principal areas, outlined in the following three paragraphs.

The first function of *Binding Selection* is to compute potential e-QCs, each consisting of bindings of l-QCs in the local domain with the o-QCs of downstream peers. The process of identifying this list of e-QCs will take into account business-related constraints (policies) when generating combinations of l-QCs and o-QCs. The list of e-QCs may also take into account simple engineering constraints, such as destination addresses that can only be reached through a single egress. This function of *Binding Selection* is essentially the function originally specified in [D1.1] for the *QC Mapping* function block.

The second function of *Binding Selection* is to select a set of e-QCs that meets the QoS requirements of the forecast inter-domain traffic and which makes optimal use of network resources both within the domain and on the inter-domain links. It achieves this by interfacing with the function block responsible for computing (and optimising) the Inter-domain configuration, *Inter-domain Resource Optimisation*, and through it the function block responsible for computing the optimal intra-domain network configuration, *Offline Intra-domain TE.*

The third function of *Binding Selection* is to command the *pSLS Ordering* function block to negotiate pSLSs with peers, and to identify ranges of parameters (such as bandwidth, one-way transit delay, cost) and groups of pSLSs which have to be ordered or negotiated.

The detailed draft specifications of the algorithms have been suppressed in the public version of this document as they are in the process of being validated. The final versions will appear in D1.3.

5.4.1.2 Objectives

The overall objectives of *Binding Selection* are to select a set of e-QCs that meet the QoS requirements of the forecast inter-domain traffic while taking into account the inter- and intra-domain configuration and traffic demands, to identify an optimal set of pSLSs that support these e-QCs, and to command the *pSLS Ordering* function block to negotiate these pSLSs.

5.4.1.3 Interface specification

Figure 47 shows the interfaces related to Binding Selection.

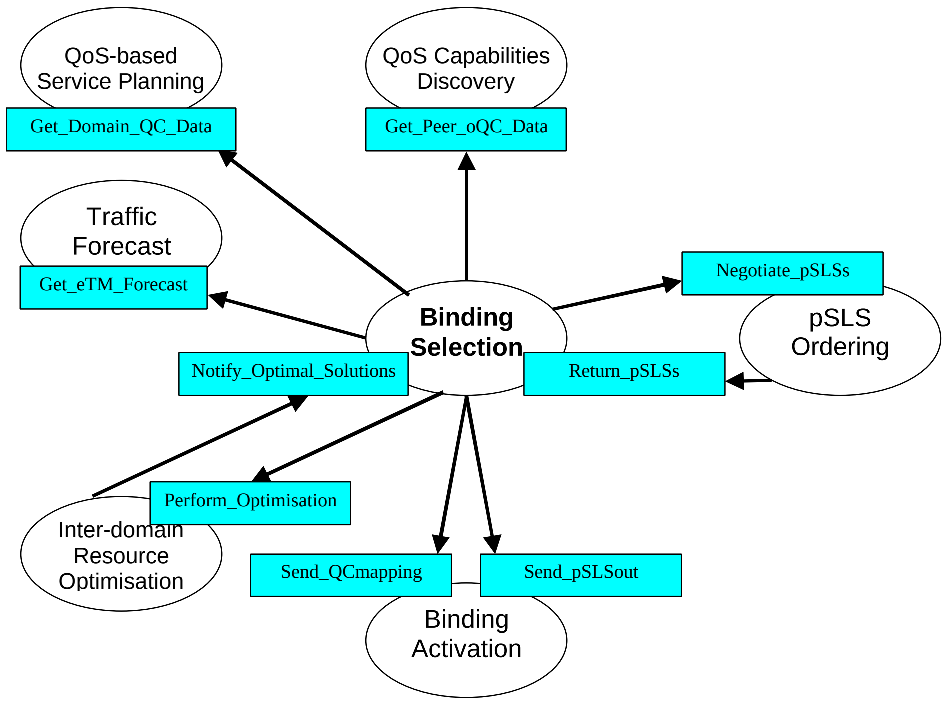


Figure 47. Binding Selection interfaces

5.4.1.3.1 From QoS Capabilities Discovery to Binding Selection

Get_Peer_oQC_Data (o-QC)

This interface provides *Binding Selection* with a list of all the o-QCs offered by peers. For each o-QC, the following information is required:

- AS identifier (or egress node interface ID);
- Destination address prefix(es);
- o-QC parameters (attribute/value pairs);
- Time schedule that the o-QC is or will be available.

5.4.1.3.2 From QoS-based Service Planning to Binding Selection

Get_Domain_QC_Data (l-QC, e-QC)

This interface provides to *Binding Selection* (a) the l-QCs that the domain provides to its customers and service peer providers within the scope of its network, and (b) a list of the e-QCs to be used to build o-QCs that the domain will offer, as defined by higher business-related activities and objectives.

For each l-QC the following information is required:

- l-QC;
- Time schedule that the l-QC is or will be available.

For each e-QC the following information is required:

- Destination address prefix(es);
- e-QC;
- Time schedule that the e-QC is or will be available.

5.4.1.3.3 From Traffic Forecast to Binding Selection

Get_eTM_Forecast (e-TM)

This interface provides the predicted traffic demand for this Resource Provisioning Cycle (in the form of the e-TM) to *Binding Selection*. The principal difference between the e-TM passed to each of *Binding Selection* and *Binding Activation* is that the former component plans for traffic flows over a longer timescale than the latter: consequently the e-TM used by *Binding Selection* is a longer-term forecast than that used by *Binding Activation*.

For each aggregate flow the following information is required:

- Ingress node ID;
- Input data rate (bandwidth);
- Destination address prefix(es);
- e-QC;
- Time schedule.

No information about committed resources is returned by *Binding Selection* to *Traffic Forecast*. This compares with *Binding Activation*, which returns an updated set of eRAM.

5.4.1.3.4 Interface with Inter-domain Resource Optimisation

This interface is bi-directional. *Binding Selection* passes to *Resource Optimisation* a specific configuration of flows including proposed l-QCs and egress node interface IDs. *Resource Optimisation* computes the best *n* inter- and intra- domain configurations, and then returns each network configuration together with its cost function value.

Perform_Optimisation (e-TM, pSLS_options, n)

The parameters passed from *Binding Selection* to *Inter-domain Resource Optimisation* are:

- Aggregate traffic flows (e-TM);
- One or more sets of options for inter-domain configuration, comprising:
  - Egress node interface ID;
  - Data rate (bandwidth);
  - Destination address prefix(es);
  - Time schedule;
  - l-QC and o-QC options;
- *n*, the number of solutions to be returned.

Notify_Optimal_Solutions (configuration, cost)

The parameters returned from *Inter-domain Resource Optimisation* to *Binding Selection* are as follows for each of the *n* solutions returned:

- The network configuration (for example, egress node interface IDs, selected l-QC and o-QCs);
- The cost of this configuration.

The difference in the two interfaces between *Binding Selection* and *Resource Optimisation* on the one hand, and between *Binding Activation* and *Resource Optimisation* on the other hand is due to one of the fundamental differences between the two binding components. *Binding Selection* is trying to define a range of "best" configurations so that it can give *pSLS Ordering* a range of pSLSs to negotiate. *Binding Activation* on the other hand operates at a shorter provisioning cycle, does not negotiate pSLSs, and uses only pre-existing pSLSs, and therefore is only interested in finding the single best configuration for its current predicted set of traffic flows.

It should be noted that in order to compute the cost of the inter-domain traffic, *Resource Optimisation* will in turn call the *Offline Intra-domain TE* function block, which requires knowledge of both inter- and intra- domain traffic.

5.4.1.3.5 Interface with pSLS Ordering

This interface is bi-directional.

Negotiate_pSLSs (candidate_pSLSs)

Binding Selection passes to *pSLS Ordering* sets of parameters; each set forms the basis of a pSLS to be negotiated with a downstream peer domain. Each negotiation may be either a new pSLS, modifications to an existing pSLS, or a cease of a pSLS. The parameters include the following, which are the principal components of a pSLS_{out} from the traffic engineering perspective:

- Egress node interface ID and/or downstream AS identifier (AS ID);
- Destination address prefix(es);
- Required data rate (bandwidth);
- Required o-QC;
- Time schedule.

The parameters may be in the form of required single values; or a range of values (e.g. min, max, mean); or qualitative measures. Values and negotiation margins of parameters may be defined.

The information passed to *pSLS Ordering* will also include logic that provides a set of negotiating parameters. Examples of negotiating logic include (a) only one of several identified pSLSs need be successfully negotiated; (b) if one pSLS is successfully negotiated with a certain set or range of parameters then parameters of other(s) are changed; (c) a list in descending order of priority list, with alternative sets of pSLSs (which can be negotiated if the initial set is not agreed by peers).

Return_pSLSs (pSLS_data)

The return value passed by *pSLS Ordering* to *Binding Selection* is a statement of which pSLSs were successfully negotiated (as new, changed or ceased) and the agreed parameter values. The return value should also include explicit statements where pSLSs failed to be negotiated, and a reason code for each such failure (examples of this might be: insufficient bandwidth available; o-QC withdrawn; revised business-level policy).

5.4.1.3.6 From Binding Selection to Binding Activation

Send_QCMapping (l-QC,o-QC)

Send_pSLSout (pSLS)

This interface provides the output of the *Binding Selection* function block to *Binding Activation*. The output consists of the following information for each aggregate flow recorded in the e-TM:

- The expected components of a pSLS_{out} (Egress node interface ID; Data rate (bandwidth); Destination address prefix(es); o-QC; Time Schedule);
- Mapping between l-QC and o-QC.

5.4.1.3.7 Additional parameters

Policies: l-QC / o-QC combination policies.

Topology information: network topology, in particular the inter-domain link data rates (bandwidth) and their QoS parameters.

5.4.2 Binding Activation

5.4.2.1 Introduction

Binding Activation is an offline component that runs at Binding Activation Cycle epochs and produces an inter-domain traffic engineering solution (i.e. the established QC-bindings which has been put in effect for inter-domain traffic) at each short time scale.

5.4.2.2 Objectives

Binding Activation has two objectives:

- To indicate the optimal resource usage of each $pSLS_{out}$ produced by the inter-domain resource optimisation functional block. The necessity of this indication is due to the optimal resource usage of each $pSLS_{out}$ may be different from that originally defined in the $pSLS_{out}$. As a result, *Binding Activation* has to inform pSLS invocation how much resources will be actually invoked or used.
- To select the best among multiple inter-domain traffic engineering solutions. This is because the inter-domain resource optimisation may produce a set of optimal inter-domain traffic engineering solutions but only one is selected. *Binding Activation* determines and selects the best solution.

The selected inter-domain TE solution is enforced through routing decisions as well as configurations of the *Traffic Conditioning and QC Enforcement* function block, e.g. configuring the egress ASBR to perform DSCP remarking for realising an inter-domain TE solution.

The detailed draft specifications of the algorithms have been suppressed in the public version of this document as they are in the process of being validated. The final versions will appear in D1.3.

5.4.2.3 Interface specification

This section describes the interaction of the *Binding Activation* function block with the others through events, messages or signals. Figure 48 shows the interfaces related with *Binding Activation*.

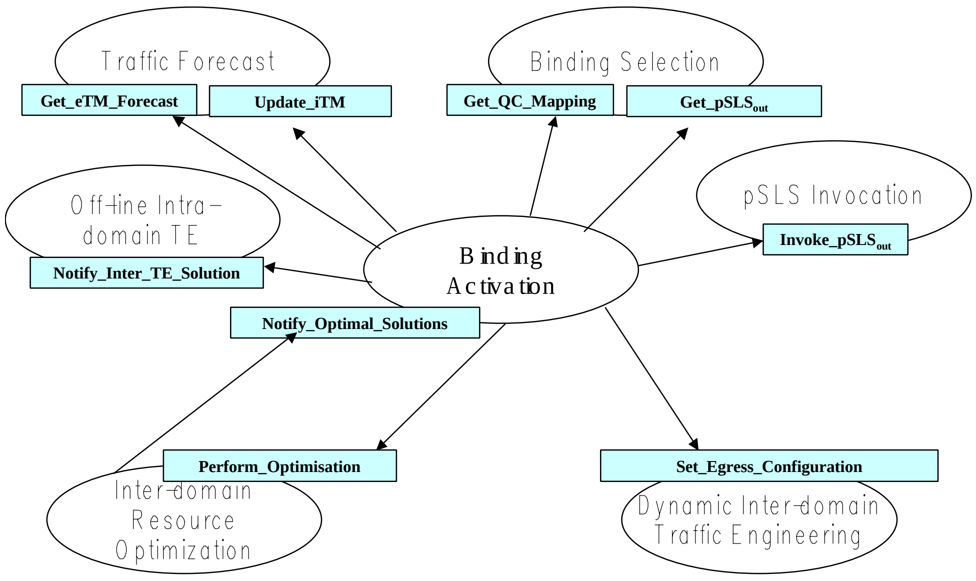


Figure 48. Binding Activation interfaces

- **Binding Activation to Traffic Forecast**

  Get_eTM_Forecast (Forecast parameters)

  This method will be called by *Binding Activation* to request and get the predicted extended traffic matrix from traffic forecast. Required traffic matrix parameters include destination prefix, a designated ingress router and requested QoS. Traffic is aggregated at each ingress router based on destination prefix.

  Update_iTM (eRAM)

  This method will be called by *Binding Activation* to pass the *Binding Activation* decision/inter-domain traffic engineering solution (i.e. eRAM) to traffic forecast for which to update the iTM.

- **Binding Activation to Binding Selection**

  Get_pSLS$_{out}$ (none)

  This method will be called by *Binding Activation* to request and get a set of pSLS$_{out}$ from Binding Selection.

  Get_QC_Mapping (none)

  This method will be called by *Binding Activation* to get a corresponding QC mapping compatibility for each received pSLS$_{out}$. The QC mapping compatibility describes a set of eligible l-QCs mappings to a specific o-QC.

- **Inter-domain Resource Optimisation to Binding Activation**

  Notify_Optimal_Solutions (eRAM(s))

  This method will be called by inter-domain *Resource Optimisation* to notify *Binding Activation* a set of optimal inter-domain traffic engineering solutions (i.e. eRAM(s)).

- **Binding Activation to Inter-domain Resource Optimisation**

  Perform_Optimisation (optimisation parameters and data)

  This method will be called by *Binding Activation* to invoke inter-domain resource optimisation by giving a set of optimisation parameters and data.

- **Binding Activation to Dynamic Inter-domain TE**

  Set_Egress_Configuration (eRAM)

  This method will be called by *Binding Activation* when the decision on which of the established QoS-bindings will be put in effect in the network for implementing e-QC has been made. The method is to pass the management directives to Dynamic Inter-domain Traffic Engineering for which to set up a configuration to realise the decision from binding activation (e.g. using BGP policies).

- **Binding Activation to pSLS Invocation**

  Invoke_pSLS$_{out}$ (eRAM)

  This method will be called by *Binding Activation* to indicate pSLS invocation functional block how much resources are estimated to be invoked or used.

- **Binding Activation to Off-line Intra-domain TE**

  Notify_Inter_TE_Solution (intra-domain configuration)

  This method will be called by *Binding Activation* to indicate offline intra-domain TE which intra-domain TE solution (i.e. the intra-domain configuration) has been selected. The purpose of this notification is to enable *Offline Intra-domain TE to* physically configure the network resources, thereby enabling the selected resource allocation.

5.4.3 Inter-domain Resource Optimisation

5.4.3.1 Objectives

Inter-domain *Resource Optimisation* computes an optimal inter-domain traffic engineering solution, taking the predicted inter-domain traffic matrix (e-TM) and intra/inter-domain resources as input. It may produce multiple optimal inter-domain TE solutions and returns the solutions to *Binding Activation* or *Binding Selection* which in turn will select and implement the best one.

The objective of inter-domain *Resource Optimisation* is to map the predicted inter-domain traffic matrix to the inter-domain network resources, satisfying QoS requirements while aiming at optimising the use of network resources within or across AS boundaries.

The detailed draft specifications of the algorithms have been suppressed in the public version of this document as they are in the process of being validated. The final versions will appear in D1.3.

5.4.3.2 Interface specification

This section describes the interaction of inter-domain *Resource Optimisation* function block with the others through events, messages or signals. Figure 49 shows the interfaces related with inter-domain *Resource Optimisation*.

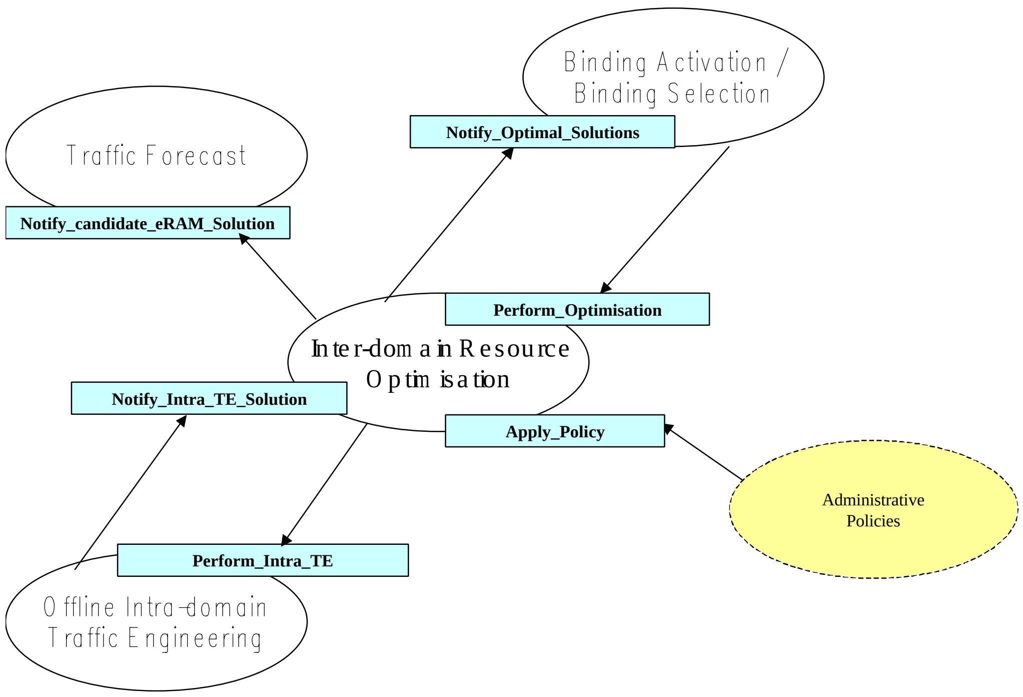

Binding Activation / Binding Selection
Notify_Optimal_Solutions
Traffic Forecast
Notify_candidate_eRAM_Solution
Perform_Optimisation
Inter-domain Resource Optimisation
Notify_Intra_TE_Solution
Apply_Policy
Administrative Policies
Perform_Intra_TE
Offline Intra-domain Traffic Engineering

Figure 49. Inter-domain Resource Optimisation interfaces

- **Binding Activation / Selection to Inter-domain Resource Optimisation**

  Perform_Optimisation (optimisation parameters and data)

  This method will be called by *Binding Activation* or *Binding Selection* to invoke inter-domain *Resource Optimisation* by giving a set of optimisation parameters and data.

- **Inter-domain Resource Optimisation to Binding Activation / Selection**

  Notify_Optimal_Solutions (eRAM(s))

  This method will be called by inter-domain *Resource Optimisation* to return to *Binding Activation* or *Binding Selection* a set of optimal inter-domain traffic engineering solutions (eRAM(s)).

- **Administrative Policies to Inter-domain Resource Optimisation**

  Apply_Policy (function + parameters)

  Any administrative policies that can affect the decision-making of inter-domain *Resource Optimisation*

- **Inter-domain Resource Optimisation to Traffic Forecast**

  Notify_Candidate_eRAM_Solution (eRAM)

  This method will be called by *Inter-domain Resource Optimisation* to notify to *Traffic Forecast* a draft or candidate eRAM configuration (*Traffic Forecast* will in turn use this to calculate a corresponding iTM and pass the iTM to *Offline Intra-domain Traffic Engineering* for calculation of cost, intra-domain resource availability and utilisation).

- **Inter-domain Resource Optimisation to Offline Intra-domain TE**

  Perform_Intra_TE (eRAM flag)

  This method will be called by *Inter-domain Resource Optimisation* to notify *Offline Intra-domain Traffic Engineering* of a request to calculate intra-domain resource availability and utilisation. *Offline Intra-domain Traffic Engineering* will use the given candidate inter-domain TE solution (eRAM) passed to *Traffic Forecast* (Note that *Offline Intra-domain TE* obtains the iTM from *Traffic Forecast*, allowing the Intra-domain algorithm to calculate the iRAM).

- **Offline Intra-domain TE to Inter-domain Resource Optimisation**

  Notify_Intra_TE_Solution (Intra domain cost F', and *intra-domain configuration)*

  This method will be called by *Offline Intra-domain Traffic Engineering* to return to inter-domain *Resource Optimisation* the intra-domain cost Φ and (optionally) the corresponding traffic engineering solution.

5.5 Dynamic Inter-domain TE

5.5.1 QoS-inferred BGP (q-BGP)

5.5.1.1 Introduction

The deployment of Internet (i.e. the best-effort Internet) was a success thanks to a fruitful cooperation between several categories of actors (service providers, standardisation bodies, regulators, manufacturers…). Service and Network Providers have deployed standard inter-domain routing protocols in order to convey reachability information between their domains. The existence of such standard protocols and interfaces has facilitated interconnection between distinct autonomous systems. Nowadays, the panorama of required information to be exchanged between Service and Network Providers and explicitly with their respective domains is different than what could be exchanged thanks to existing inter-domain routing protocols. From this perspective, it is obvious that Service and Network Providers have to evolve/update the protocols (not necessary change the core of the operational mode of these protocols but exploits extendibility capabilities of existing protocols) they are used to deploy in their ASs in order to meet the new requirements and then to be able to offer new sophisticated added value services.

QoS delivery services are seen as a part of future Internet services (*see [IAB], the IAB has qualified these services as critical*). In order to offer such services, network infrastructures (network devices capabilities, protocols, management tools, etc.) must be updated to offer this type of services. Especially and from a control plane angle, modifications need to be brought to existing signalling and routing protocols. From an inter-domain view, this is critical since Service and Network Providers should deploy means to convey QoS-related information between their domains so as QoS-based services could have a world-wide scope and then be accessible for a large set of customers in the world *(the notion of "world" doesn't mean geographical location but the affiliation to any Service and Network provider)*. This could be considered as an important risk since Service and Network Providers have to ensure backward compatibility with existing protocols. This risk should be considered carefully when designing a solution claiming to meet this backward compatibility requirement.

In this section, we describe a proposal that aims at exchanging QoS-related information between adjacent ASs. The QoS-related information exchange occurs either at the service level or at the routing level. The place this exchange occurs depends on the inter-domain QoS delivery solutions, which is deployed. Therefore, two groups of QoS delivery solutions have been identified. Hereafter the two groups:

- The first group of solutions requires propagating only an identifier that has been agreed during the pSLS negotiation phase. Of course additional QoS performance characteristics were negotiated but not exchanged in the routing level. In the rest of this document this will be denoted by *group-1*.
- The second group of solutions requires the propagation of a set of QoS performance characteristics thanks to an inter-domain routing protocol associated with an identifier. The nature of the QoS-related information to be exchanged has been agreed in the pSLS negotiation phase. In the rest of this document this will be denoted by *group-2*.

In addition, the proposal - That benefits from the extensibility capability offered by the Border Gateway Protocol (BGP) [RFC1771]- meets a set of generic requirements that are described below. The aforementioned proposal could apply for any kind of inter-domain QoS delivery solution that is based on an exchange of QoS-related information between domains. In particular, within the context of MESCAL, the proposal meets the "solution options"-specific requirements.

The MESCAL solution options requirements on QoS-related information will be studied in details and then each solution option will be classified according to the aforementioned grouping.

This section is organised as follows: Sub-section 3 lists goals and needs of a means allowing exchange of QoS-related information. Sub-section 4 details the MESCAL solution options requirements and the nature of required QoS-related information to be carried in q-BGP messages. Sub-section 5 presents q-BGP specifications in terms of messages and route selection process.

5.5.1.2 Definitions

Within this section the following terms are used as defined below:

- QoS-related information can be expressed in terms of one-way delay, inter-packet delay variation, loss rate, DSCP marking, or a combination of these parameters.
- 'QoS service'-related attributes: denotes dedicated q-BGP attributes for the usage of a given QoS service.
- Inter-domain QoS delivery solution is used to denote an inter-domain system that aims at offering QoS services.

5.5.1.3 Objectives and Needs

Big ISPs use the Border Gateway Protocol (BGP) protocol in order to interconnect their ASs with the ones of their peers. It is the unique inter-domain routing protocol that is used in the Internet. More and more have gained the experience of configuring and understanding the operational mode of this routing protocol. Thus, several proposals aiming to enhance the capabilities of BGP to carry more information than what have been included in the BGP specifications have been proposed. The goal of this section is to describe how BGP could be used as a means to convey QoS-related information between adjacent autonomous systems especially within the context of MESCAL while taking into account the three solution options detailed in [D1.1] (note that the solution is generic and could be applied to all kind of inter-domain QoS delivery that is based on exchanging QoS-related information).

These solution options rely on an exchange of QoS-related information that takes place between adjacent ASs (also called service peers). This exchange occurs at the service level (management plane) and at the routing level. It consists in negotiating QoS guarantees during the pSLS negotiation phase and then in propagating them (or part of them) thanks to an inter-domain routing protocol. The means of exchanging QoS-related information should meet a set of generic requirements. It should be dynamic, scalable and should be able to propagate topology changes without any significant impact on the existing best-effort based network infrastructure. This document aims at clarifying these requirements and examine if they are applicable for all the solution options since each MESCAL solution option could require specific QoS messages and route selection process.

This document doesn't intend to detail the BGP protocol specification or its operational mode. For more information about BGP, the reader can refer to [RFC1771] and other related IDR working group [IDR] RFCs. This document will focus on QoS-related information that needs to be conveyed by BGP messages and the use of this information by the route selection process.

5.5.1.4 Towards a QoS-inferred BGP

The purpose of this section is to identify the specific requirements of the three solution options described in [D1.1] in order to convey relevant QoS information between autonomous systems. In addition, and from a routing/signalling perspective, we identify additional requirements that are necessary for each solution option to become operational. From BGP standpoint, the intent is to identify (1) the possible lacks (2) the information to be carried in the BGP messages and (3) required modifications in order to meet the solution options requirements. These requirements will be taken into account when designing a solution that will apply for any inter-domain QoS delivery solution that is mainly based on an exchange of QoS-related information between service providers' domains. In other words, the purpose is to identify the group (See Introduction) each solution belongs to.

In the rest of this document, the resulting modified BGP will be denoted by q-BGP (for QoS inferred BGP). Both the route selection process and BGP attributes will be considered.

5.5.1.4.1 Analysis of the three solution options needs

5.5.1.4.1.1 The loose solution option

5.5.1.4.1.1.1 The loose solution option assumptions

This solution option relies deeply on the use of both q-BGP protocol and the meta-QoS-class concept. The resulting QoS-Internet can be viewed as a set of parallel meta-QoS-class planes running distinct instances of an inter-domain routing protocol.

When a service agreement exists, the service peers exchange (at the service level) QoS information about their reachability scopes. This exchange is achieved on a per meta-QoS-class basis. These agreements impact the routing policy filters and grant the remote service peer to benefit from its neighbour's inter-domain QoS capabilities (note that appropriate policies could be negotiated, for example to restrict the set of authorised destinations…).

Reminder:

- *The prior establishment of pSLSs conditions the exchange of inter-domain connectivity information per meta-QoS-class.*
- *Each meta-QoS-class is identified by a well-known identifier*
- *Each AS achieves a DSCP swapping operation: the egress AS changes its l-QC DSCP into the agreed Meta -QoS-Class DSCP and the ingress AS point changes the meta-QoS-Class identifier into its l-QC DSCP)*
- *Each AS announces to its service peers the network prefixes that can be reached within each meta-QoS-class plane*

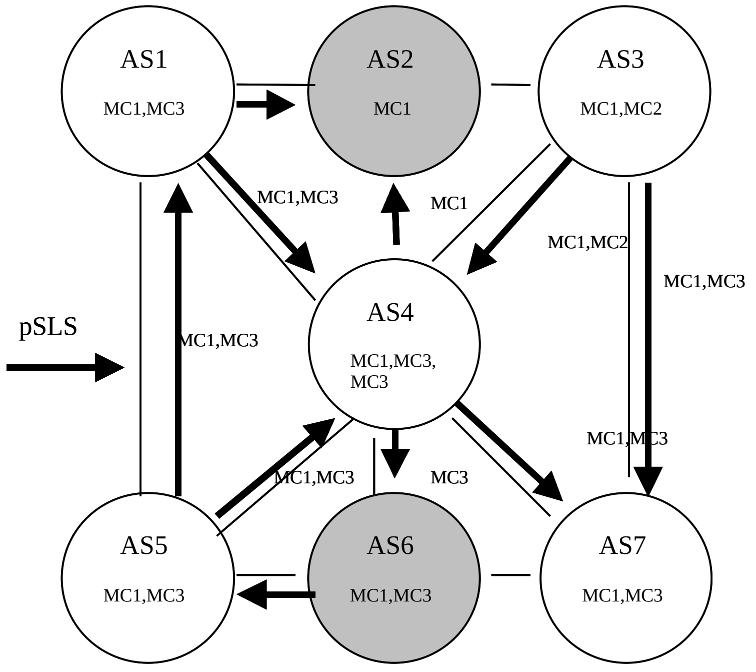


Figure 50. Reachability information exchange *a la* loose solution option

Consequently, BGP UPDATE messages must include a meta-QoS-class identifier, so that each message can be processed within the context of the corresponding meta-QoS-class plane.

Since inter-domain paths depend on the meta-QoS-class used to signal the requested quality of service guarantees, it becomes necessary to store the information associated with an individual update in a Routing Information Base (RIB) instance dedicated to the meta-QoS-class the update is intended to. Handling as many Routing Information Bases (RIB) as available meta-QoS-classes also requires that a route selection process runs for each RIB instance in order to compute the resulting Forwarding information Bases (FIB).

When transiting through a set of ASs, the QoS treatment experienced by a datagram is "*consistent*" in all transited ASs (note that "*consistent*" denotes the fact that the treatment received by the IP packets in each AS conforms to the corresponding meta-QoS-class definition. It doesn't mean that the QoS characteristics applied to the datagrams when crossing the different ASs are the same). From this perspective, conveying a meta-QoS-class identifier in q-BGP announcements could be sufficient to learn end-to-end QoS paths. In this case, the BGP route selection process could be kept unchanged but it would not select an optimal path since QoS characteristics resulting from the concatenation of each l-QC encountered along the path would not be present and thus not considered by the selection process. If this information was inserted in q-BGP update messages it could be advantageously taken into account by the q-BGP route selection process to select an optimal path. This indeed would enable to tune more precisely the route selection process in order to select routes according to more sophisticated routing policies.

QoS-related information inserted in q-BGP update messages is intended to facilitate the selection of the best possible end-to-end route. But this information could be of different nature. It could be administratively enforced. In that case it would not change too frequently. Or, it could be much more dynamic (result of a measure for instance) and in that case the frequency of changes would be much more higher.

Three scenarios need to be studied:

- q-BGP carries only meta-QoS-class identifiers
- q-BGP carries meta-QoS-class identifiers AND end-to-end QoS information
    - QoS information are administratively enforced
    - QoS information is dynamic and reflect the real status of the network

5.5.1.4.1.1.2 Only MC identifier is available

***Assumption**:* BGP messages carry only meta-QoS-class identifiers.

***Impact on BGP**:* In this context, the route selection process remains the native BGP one. The main selection criterion is still the AS path length. Note that a route selection is achieved for all destinations of each available meta-QoS-class plane.

***Example**:* Let's consider the example of Figure 51:

Thanks to the pSLSs established between the different domains involved in this example, AS6 can reach prefixes located in AS2 within MC1 plane thanks to several paths:

1. AS5, AS1
2. AS5, AS4
3. AS5, AS1, AS4

AS6 can choose either the first or the second path since those ones are the shortest (i.e. a smaller AS_PATH attribute). The final selection will be based on the local routing policies enforced by AS2.

***Summary**:* The characteristics that can be put forward for this situation are:

- No modification of the BGP route selection process.
- In normal exploitation conditions, the selected paths are guaranteed to be in the corresponding meta-QoS-class plane.
- The route selection process doesn't necessarily select the optimal path.
- In case of problem (l-QC over-charged in an AS along the path leading to a meta-QoS-class criterion break) there is no way to detect and correct the failure.

Operational mode:

- *Each service peer adds a meta-QoS-Class identifier to its BGP announcements in order to identify the meta-QoS-class plane the announcements is intended to (this information is represented by MCi)*
- *Upon reception of these q -BGP announcements, the receiving AS (listening ASBR) proceeds to the classification of each announcement and queues it in the related RIB*
- *In this case, the per meta -QoS-class plane route selection decision -making remains identical to the classical route selection process*

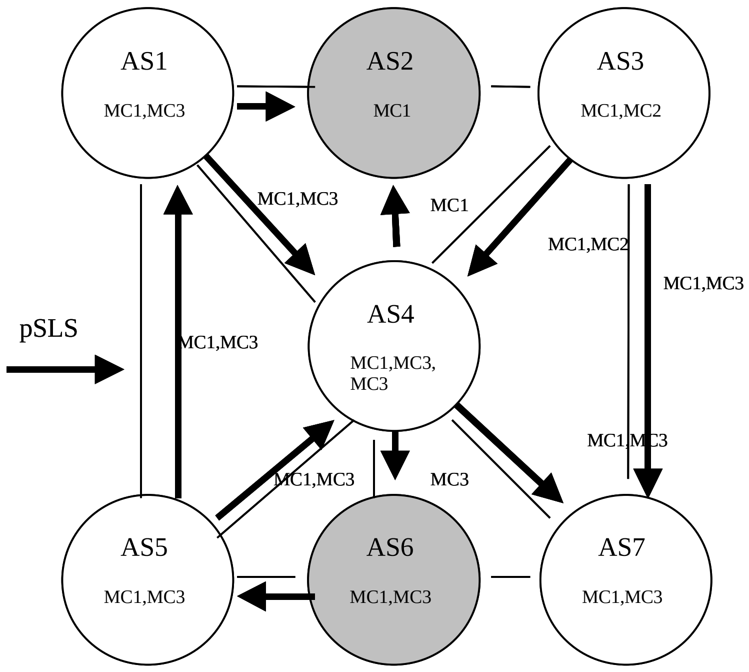


Figure 51. Only meta-QoS-class identifiers are carried in the q-BGP messages.

5.5.1.4.1.1.3 Administrative QoS-related information is available

***Assumption**:* q-BGP carries meta-QoS-class identifiers AND end-to-end administrative QoS information. This QoS information is the concatenation of the characteristics of the l-QCs experienced along the AS path.

The assumption of 5.5.1.4.1.1.1 applies here, i.e. QoS information must be exploited to perform route selection. This enables administrators to define precise policies that will lead to the selection of the best route (according to administrators' criteria). This leads to a significant modification of the route selection process, as it must now take into account the QoS information to choose the best route, depending on the policy enforced by the administrator per meta-QoS-class.

In the example illustrated by Figure 52, thanks to pSLSs established between the different domains involved, AS3 can reach prefixes located in AS7 within meta-QoS-class MC1 plane thanks to several paths that are explicitly:

- e-QC137
- e-QC134

The AS7 has to decide which path to activate. This is done by comparing the two e-QCs and choosing the best one (thanks to a well-know or proprietary QC comparison logic)

Operational mode:

- *e-QCijk refers to end-to-end QoS characteristics announced by ASk to ASj in the MCi meta-QoS-class plane.*
- *For each network prefix announcement, the AS must associate a meta-QoS-class identifier AND an e -QC parameter. The e -QC parameters result from the concatenation of the QoS performance characteristics of l-QC of the ASs traversed by the AS path. This is represented in the figure by {MCi, e-QCijk}*
- *Upon the reception of q -BGP announcements, each AS computes the resulting e -QC parameters in concatenating the QoS performance characteristics of its l-QC with those of the received eQC.*
- *The route selection process mainly consists in selecting an inter - domain path that optimizes end-to-end QoS characteristics of the route for the meta-QoS-class, which is considered.*

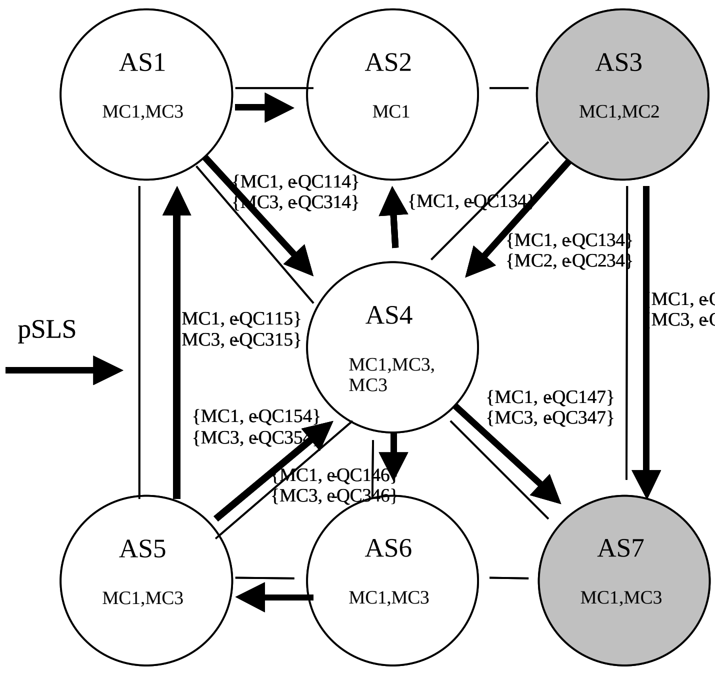


Figure 52. Use of meta-QoS-class identifier and end-to-end QoS characteristics.

The characteristics of this approach are:

- In normal exploitation conditions, the selected paths are guaranteed to be in the corresponding meta-QoS-class plane.
- Modifications to the route selection process are quite important.
- Selection of an optimal QoS route can been achieved with more accuracy.
- In case of problem (l-QC over-charged in an AS along the path leading to a meta-QoS-class criterion break) there is no way to detect and correct the failure.

We can observe that, if QoS information is administratively enforced, the route selection process will always make the same decision in normal operation conditions. This observation points out that, most of the time, conveying administratively enforced QoS information will over-charge the network, as it won't be exploited for any update. This can lead to a discussion on the real interest of conveying such administratively enforced QoS information:

- If we consider the main goal of loose solution option, which is to select and guarantee a path within a meta-QoS-class plane, conveying administratively enforced QoS information has no added value. Indeed, in normal operational conditions, the selected path will always be the same; in case of problem, there is no mean to detect that the meta-QoS-class plane guarantee is fooled. Therefore, we could consider that conveying static QoS information is useless in so far as it doesn't bring more than the classical route selection process based on AS path. Moreover, this brings more complexity into BGP for a relatively poor added value.

- If we want to extend the main principle of loose solution option and allow the selection of the best route (depending on predefined criteria, like the best delay, the lower jitter…) then conveying static QoS information is justified. Additionally, even if the route selection process would not use these QoS information, it could be used by network administrator prior pSLS establishment or by customers for an informational purpose. It should also be noted that QoS performance characteristics of a route could change following traffic-engineering tasks achieved in remote ASs leading so the route selection process to select another route.

5.5.1.4.1.1.4 Dynamic QoS information is available

The assumptions are similar to those of the former scenario except that concerning QoS-related information that becomes dynamic and results of an active measurement protocol. The principles mentioned in the previous section also apply here regarding the route selection process and the ability for administrators to apply specific policies.

Operational mode:
- *e-QCijk refers to end-to-end QoS characteristics announced by ASk to ASj in the MCi meta-QoS-class plane.*
- *For each network prefix announcement, the AS must associate a meta-QoS-class identifier AND an e-QC parameter. The e-QC parameters result from the concatenation of the QoS performance characteristics of l-QC of the ASs traversed by the AS path. This is represented in the figure by {MCi, e-QCijk}*
- *Each AS runs an active measurement protocol that provides the real-time QoS performances parameters of its own l-QCs*
- *Upon the reception of q-BGP announcements, each AS computes the resulting e-QC parameters in concatenating the measured QoS performance characteristics of its l-QC with those of the received e-QC.*
- *The route selection process mainly consists in selecting an inter-domain path that optimizes end-to-end QoS characteristics of the route for the meta-QoS-class, which is considered.*

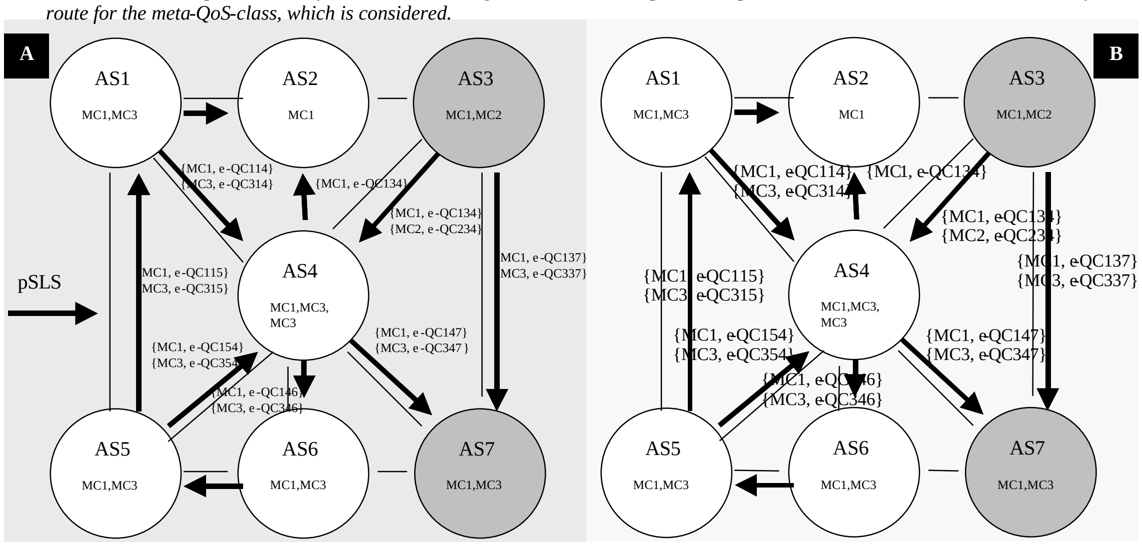


Figure 53. Use of meta-QoS-class identifier and dynamic end-to-end QoS characteristics

Let's consider the following example (Figure 53) where AS3 wants to join prefixes located in AS7 within the meta-QoS-class MC1. In case "A" we suppose that e-QC134 is better than e-QC137, then the route that will be chosen is that which follows the e-QC134. In the case "B" we suppose that e-QC137 is better than e-QC134, then the route that will be chosen is that which follows the e-QC137 since it is now better than e-QC134. The changes of the e-QC parameter's values could be frequent. Therefore, under the same configuration, different routes could be chosen depending on results of e-QC comparison logic.

The fact that the QoS-related information is regularly updated provides an important advantage compared to the two other solutions. Indeed, it allows the detection of a break in the meta-QoS-class plane guarantee paradigm. Therefore, the route selection process can perform another choice that will ensure the traffic will still be forwarded in the required meta-QoS-class plane. Thus, conveying dynamic QoS information brings a real advantage, which is not present with administratively enforced QoS information. Nevertheless, the updates will be more dynamic and will impact the convergence of the BGP and the stability of the routing tables.

The characteristics of this scenario are:

- In normal exploitation conditions, the selected paths are guaranteed to be in the corresponding meta-QoS-class plane.
- Modifications to the route selection process are quite important.
- Possibility to tune the policy, which will lead to the best route selection.
- In case of problem (l-QC over-charged in an AS along the path leading to a meta-QoS-class criterion break) the regular update of the QoS information enables to select another path that will remain in the required meta-QoS-class plane.
- This approach can generate very harmful oscillations being able to seriously harm the stability of the QoS Internet.

5.5.1.4.1.1.5 Summary

Table 9 can be considered as the set of recommendations applying to the different contexts and objectives.

Attributes conveyed in BGP	***Features***	***Impact on BGP***
MC	• Ensures that route remains in the same meta-QoS-class plane. • Doesn't compute an optimal path with regards to end-to-end QoS performance characteristics	• No impact on classical route selection process • Slight modifications to BGP protocol • Modification of the information contained in the RIB (that becomes q-RIB) • Duplication of routing process and associated RIBs (one q-RIB per meta-QoS-class)
MC and administrative QoS information	• Ensures that route remains in the same meta-QoS-class plane. • Compute an optimal path with regards to end-to-end QoS performance characteristics	• Modification of the information contained in the RIB (that becomes q-RIB). • Duplication of routing process and associated RIBs (one q-RIB per meta-QoS-class). • Modifications to BGP protocol. New attributes have to be defined. • Impact on classical route selection process. Route selection process relies on QoS information
MC and dynamic QoS information	• Ensures that route remains in the same meta-QoS-class plane. • Compute an optimal path with regards to end-to-end QoS performance characteristics • Detect an overloading of a given l-QC.	• Duplication of routing process and associated RIBs (one q-RIB per meta-QoS-class). • Modifications to BGP protocol. New attributes have to be defined. • Impact on classical route selection process. Route selection process relies on QoS information

Table 9. Summary of the loose solution option recommendations and requirements

As a result of this analysis, MESCAL has decided to adopt the second scenario that aims at announcing both meta-QoS-class identifiers together with administrative QoS performance characteristics.

5.5.1.4.1.2 The statistical Solution option

Within the context of the statistical solution option, the use of q-BGP can be considered as optional. Indeed, each pSLS contains an exhaustive description of the destination prefixes attached together with their associated QoS performance characteristics and guarantees. The management system of each AS stores this information. Remote destination network prefixes are known as of the pSLS negotiation phase and before their effective activation. Consequently, knowledge of potential next hop ASs for a given destination (and then the choice of a path to a given destination) can be directly deduced from pSLS information maintained by the management plane. Considering that a routing protocol generally achieves two main elementary functions, which are: routes discovery and route selection it can be concluded that q-BGP (for this solution option) doesn't bring any real added-value to the inter-domain routes discovery function since those routes are already known and enforced by the management plane of the peering ASs. Then, inter-domain *routing discovery could be completely management-based* since pSLSs include routing-related information.

Nevertheless q-BGP can be advantageously used as a means to enhance the effectiveness of this solution option especially by:

- Propagating inter-domain routes within the domain, thanks to q-iBGP
- Signalling dynamically the effective availability of the routes
- Providing a means to select dynamically alternative routes in case of failure: this requires establishment of multiple pSLSs allowing reaching the same destinations with similar o-QC.
- Making potentially easier the deployment of some load balancing features between paths that serve the same destination(s) and in which traffic will experience similar o-QC. Note that this latter point will not be developed and will be kept for further studies.

Within the scope of this solution option, routing and forwarding is DSCP based. When q-eBGP is activated between two peering domains, q-eBGP updates must convey, in addition to the network prefix, a DSCP value, which indicates to the upstream AS, the DSCP value to use in order to benefit from the QoS performance guarantees attached to the o-qc a given destination belongs to.

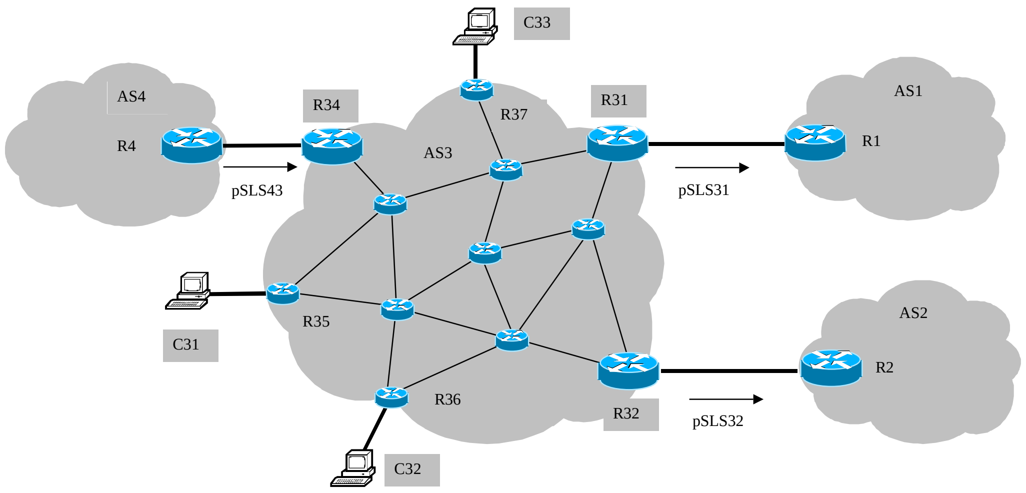


Figure 54. The statistical solution option operational mode

In Figure 54, AS3 can deploy a maximum of N local-QoS-classes quoted l-qc-i where i can vary between 1 and N (N<64). In order to benefit of a particular QoS treatment for its traffic, a customer attached to AS3 signals the requested QoS using the appropriate DSCP value. This DS code point is bound to an appropriate AS3's l-qc. Note that several DSCP values can be bound to the same l-qc in order to solve the QC splitting problem.

Let's assume now that AS3 buys the same destination prefix "D" from AS2 and AS1 with some QoS guarantees. Thanks to pSLS31 and pSLS32 In this example QoS guarantees are supposed to be almost the same and AS3 decides to bind o-qc31 and o-qc32 with one of its own l-qcs: l-qc1. As a consequence of these two bindings two new e-QCs are now available. Their respective QoS characteristics are very close:

- e-qc-1-31=l-qc-1 ⊕ o-qc-31
- e-qc-1-32=l-qc-1 ⊕ o-qc-32

From a commercial perspective, they could be sold as a same o-qc: o-qc1-31-32

Once pSLSs have been activated, AS1 and AS2 send q-eBGP updates indicating that destination D is available and should be signalled using a particular inter-domain DSCP value. In the example above these values are DSCP31 for AS1 and DSCP32 for AS2. The prefix D is announced to all AS3 routers. Thus, R31 and R32 send q-iBGP updates to all AS3 routers. These updates are only relevant for the DSCP routing plane corresponding to l-qc-1. Each update received from a downstream AS must consequently be interpreted by the ASBR. R31 will have to consider, for each announcement, the network prefix, the associated inter-domain DSCP value and the identity of the q-eBGP speaker. Using this information, it can retrieve from the management plane, the o-qc this announcement belongs to and gets in return the bindings details. In our case, these binding details will indicate to the two ASBR that q-iBGP update for destination D must be done within DSCP1 plane. Learned DSCP value must be accordingly swapped to its new intra-domain value.

As a consequence of this processing, R35, R36 and R37 have the same q-RIB information concerning destination D and have now to select a route toward this destination.

Destination	*DSCP plane*	*q-BGP Next Hop*
D	DSCP-1	R31
D	DSCP-1	R32

Table 10. A very simplified q-RIB example

All these paths are equivalent in term of QoS performance. For an AS3 customer, there is no difference joining D via R31 or R32. Indeed, each router can select any of these routes since, by construction, they all provide the same QoS performance guarantees. But, in the above example R36 would certainly select R32, and R37 would select R31, if the network provider would apply a "hot potato" policy.

As a consequence q-BGP doesn't need that QoS information be carried in updates for feeding the route selection process. The decision to announce a destination within a DSCP plane is enforced by the binding process. This is a 100% administrative decision. Resulting QoS characteristics of each o-QC is perfectly known, stored and maintained by the management plane. It doesn't change if the pSLS remains unchanged.

The network provider (via the management plane) should ensure that it never propagates in q-iBGP a prefix that would be far (in terms of QoS characteristics) from the other prefixes already injected for the same destination(s). In other terms, all announcements, within a given DSCP plane, concerning a given destination D, must be seen as a "similar to" o-qc even if they are really achieved with different remote o-qcs. It is the responsibility of the management plane to ensure this consistency. But this is more a binding than a q-BGP constraint.

From this perspective, the per-DSCP plane route selection process remains unchanged compared to the existing BGP selection process.

In addition, in order to illustrate the complexity of managing pSLSs that could leads definition of services to reach a given destination with similar QoS performance characteristics, let's consider the example of Figure 55:

Operational mode:

- *Exchange of traffic that will benefit from o-QC is conditioned by the establishment of appropriate pSLSs*
- *A pSLS include o-QC performance guarantees and a list of prefixes*

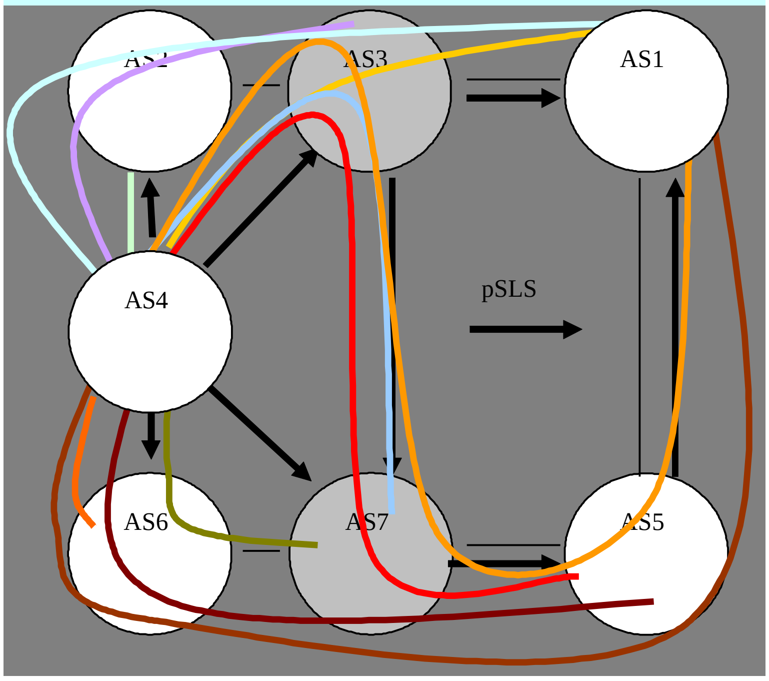


Figure 55. The statistical solution option operational mode-bis

AS4 has to manage at least 11 o-QCs that should be differentiated in order to achieve advanced task like load balancing. Some issues are to be solved like the insufficiency of DSCP range that could be used. Some ideas could be put forward as the use of an o-QC identifier that will identify a given o-QC obtained thanks to a pSLS negotiation.

From this perspective, it is obvious that the route selection process for the statistical solution option is less complex than the loose solution option one and minor modifications are to be added to BGP. Nevertheless, offline traffic engineering functionalities *are* complex and have to provide the dynamic inter-domain routing with (in a static or automatic fashion):

- The configuration of routing policies
- The configuration of LOCAL_PREF
- The configuration of the load balancing-related policies: between pSLS, between similar o-QC, …

Off-line TE functions:

- Configure filters
- Configure LOCAL_PREF
- Configure load-balancing related policies (ratio,..)

Dynamic inter-domain TE functions/protocols:

- Apply routing policies that have been configured by the off-line TE
- Learn/announce reachability information
- Select the best route based on the configured policies

Figure 56. Interaction between off-line TE and dynamic inter-domain TE.

As a conclusion, it can be stated that in order to be able to support the statistical solution option, q-BGP:

- Must associate to each update the DSCP value corresponding to the agreed o-qc the announcement belongs to.
- Must run one routing decision process per DSCP plane.
- Can keep the standard routing decision algorithm at least as long as load-balancing and dynamic inter-domain bandwidth constraints are not considered.

5.5.1.4.1.3 The hard solution option

The hard solution option exploits q-BGP announcements as a means to detect IP address of end-points in distant domains in order to build end-to-end LSPs. [D1.1] deliverable specifies that the hard solution option uses q-BGP announcements in order to learn new destinations per meta-QoS-class. However in Section 2.5, new solutions have been proposed in order to decrease the size of the routing table when q-BGP is used as a means to convey QoS-related information.

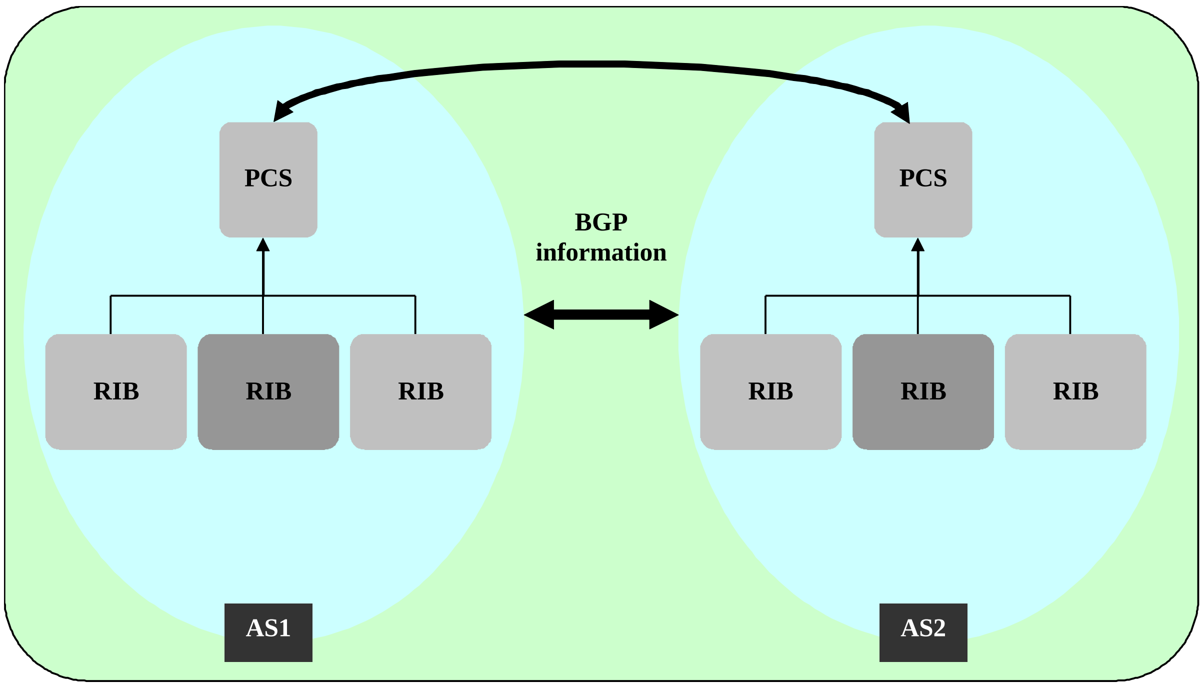


Figure 57. PCS communication.

The two proposals were introduced in order to solve issues resulting of the inter-working between these two solution options:

- The single signalling channel: in this proposal, the same q-BGP announcements are used by the loose and the hard solution option. The activation of the hard service option at a given peering point is conditioned by the activation of the loose service option. Route filtering is common for the two solution options. Hard solution option holes encountered along an inter-domain path are signalled. This information is ignored by the loose solution option but is taken into account by the PCS of each domain in order to compute an inter-domain LSP. Under these conditions, q-BGP behaviour is the same for the two solution options and the requirements stated in 5.5.1.4.1 are applicable in the context of the hard solution option also.
- The double signalling channels: in this proposal, we introduce a mechanism to distinguish q-BGP announcements of each solution option. A solution option identifier inserted in q-BGP updates achieves this. At a given peering point, the hard and the loose service options can be activated independently of each other. This discrimination doesn't induce a difference on q-BGP behaviour but only indicates to which solution option those announcements are intended for. Hard solution option specific information can differ according to two variants which are discussed below:

- Announcement of potential LSP termination end-point addresses (routers' loopbacks or interfaces) are: in this case the announcement only differs by the value of the solution option identifier. Hard solution option specific information is ignored by the loose option. The same rules than for the loose option apply for building those announcements. External hard solution option routes are not taken into account when feeding the q-FIB. Only, intra-domain interface and loopback addresses of the routers must be present in the q-FIB. Then, the q-BGP behaviour in the hard solution option is the same as for the loose one. From this angle, the requirements stated in 5.5.1.4.1 apply also in this case.
- Announcement of only the PCS addresses: in this case, we decrease the number of q-BGP announcements that are reduced to one announcement per PCS per AS. This case is similar to the previous one since these announcements are still differentiated from the loose ones. The requirements stated in 5.5.1.4.1 apply also in this case. Note that in this case the related information isn't stored in the FIBs. This particularity should be take into account in the implementation phase when considering forwarding aspects.

5.5.1.4.2 Towards a q-BGP convergent solution

5.5.1.4.2.1 q-BGP behaviours

The q-BGP protocol (abstraction made of behavioural aspects) should, as far as possible, be able to operate independently of a given inter-domain QoS delivery solution. Thus it should be able to support all kind of solutions based on an exchange of QoS-related information. Within the MESCAL context, q-BGP must more specially meet the requirements of group-1 and group-2 solutions (as defined in the Introduction). q-BGP should then be unique but could have distinct behaviours depending on the requirements and goals of the solutions groups.

A q-BGP behaviour depends deeply on the nature of the QoS-related information carried by its messages. If q-BGP messages carry only a QC identifier (this identifier could be a DCSP code-point or a proprietary identifier), offline traffic engineering functions are certainly complex but the q-BGP route selection process complexity is reduced. This complexity increases when a set of QoS characteristics are associated with each QC identifier. The route selection process can use either the QC-identifier for all solutions that take part of group-1 or the QC-identifier and QoS performance characteristics for solutions belonging to group-2. Figure 58 summarises these behaviours:

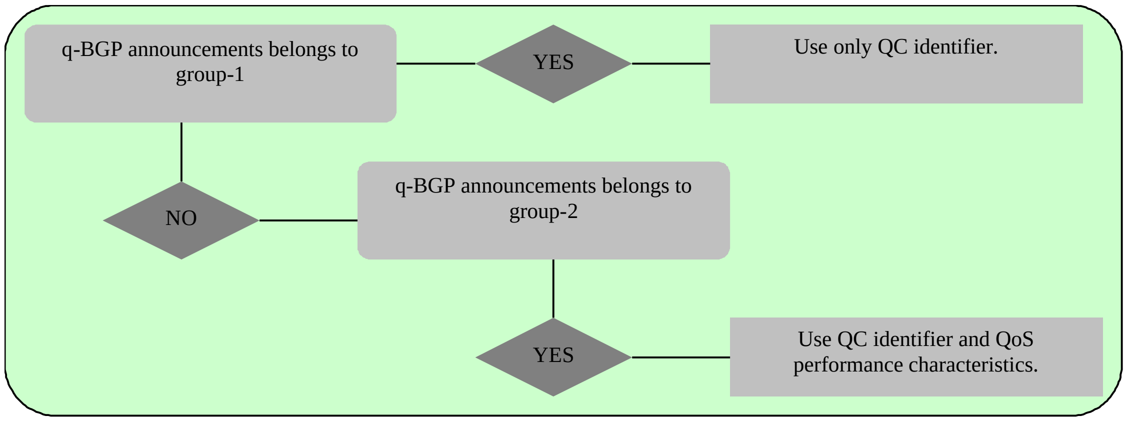

q-BGP announcements belongs to group-1
YES
Use only QC identifier.
NO
q-BGP announcements belongs to group-2
YES
Use QC identifier and QoS performance characteristics.

Figure 58. Route selection process required information per group.

From this standpoint, q-BGP protocol should be able to detect the group it serves. Then, it is required to introduce an additional step in the above diagram consisting at exchanging QoS service capabilities supported by each AS (BGP speaker). Therefore, Figure 58 becomes as shown in Figure 59:

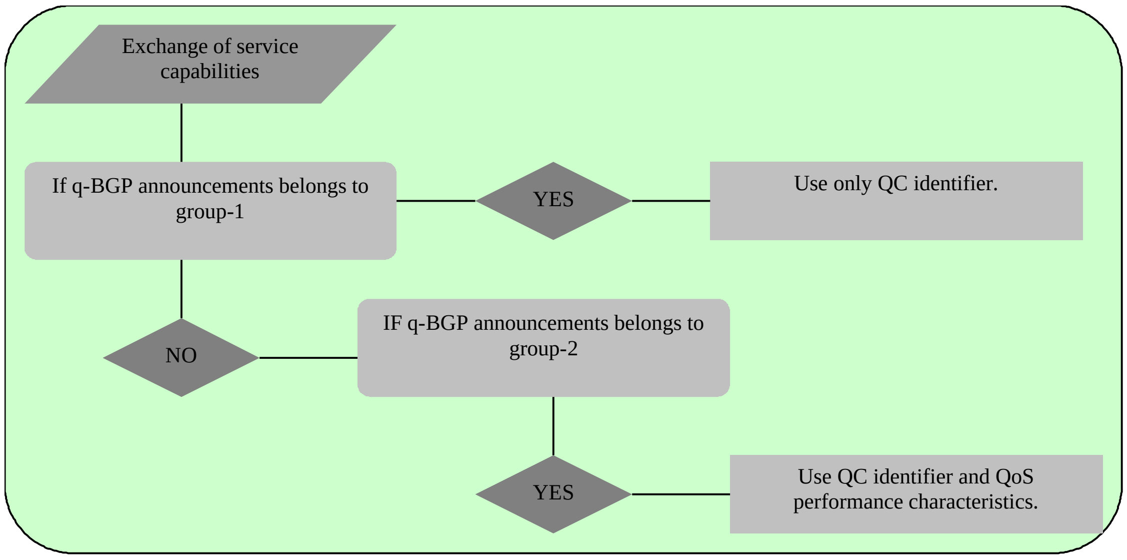

Exchange of service capabilities
If q-BGP announcements belongs to group-1
YES
Use only QC identifier.
NO
IF q-BGP announcements belongs to group-2
YES
Use QC identifier and QoS performance characteristics.

Figure 59. Towards convergent q-BGP-bis.

The purpose of the first step of the above diagram is to enable q-BGP peers to exchange the QoS service capabilities they support. Thus, neighbours q-BGP peer can ensure quickly that they are able to understand each other, thus avoiding inopportune BGP closing of session.

5.5.1.4.2.2 Applicability to MESCAL solution options

The above discussions (see sub-section 5.5.1.4.1) have revealed some common issues, related to the exchange of QoS information, between the loose and the hard solution options that q-BGP could solve in a similar manner. In particular, q-BGP messages should carry a meta-QoS-class identifier together with the QoS performance characteristics associated to the destination network prefix. However, in the case of the statistical solution option, q-BGP should carry only a DSCP identifier.

This could be summarised in Figure 60:

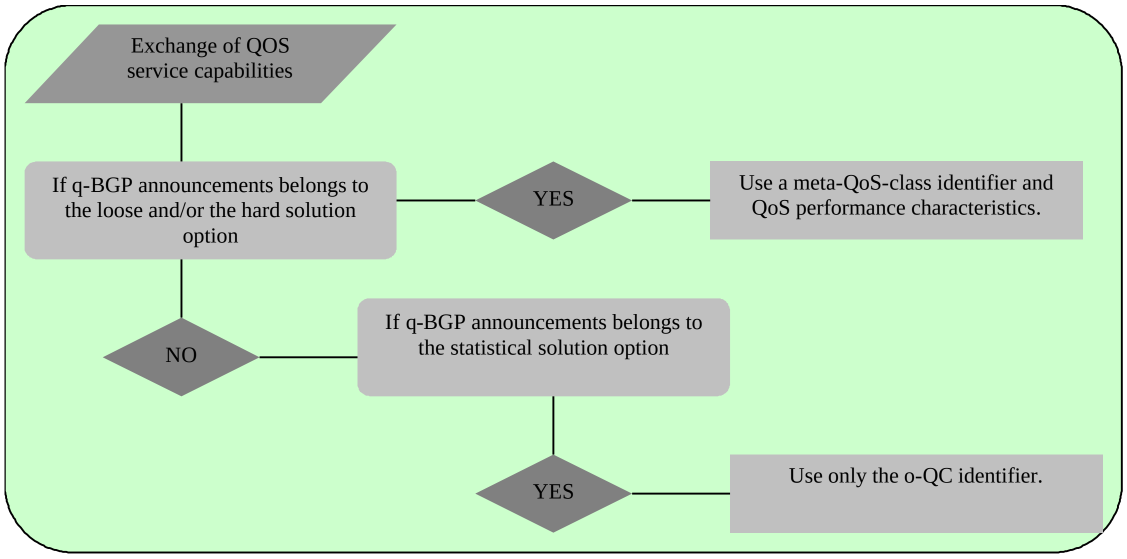

Exchange of QOS service capabilities
If q-BGP announcements belongs to the loose and/or the hard solution option
YES
Use a meta-QoS-class identifier and QoS performance characteristics.
NO
If q-BGP announcements belongs to the statistical solution option
YES
Use only the o-QC identifier.

Figure 60. q-BGP in case of MESCAL solution options.

5.5.1.5 q-BGP specification

The discussions above have shown that q-BGP needs to carry some pertinent information according to the group it serves. This information is listed below:

- QoS Service Capabilities: this is motivated by the fact that peering entities need to ensure each other of their QoS service capabilities in order to avoid peering disruptions when a new service option is activated. q-BGP have to indicate the solution group(s) it can serve and thus indicating what kind of information can potentially be carried by its messages. This will be achieved thanks to the capability optional attribute defined in [RFC3392].

- QC identifier: This identifier will be used to differentiate the extended QCs (i.e. between meta-QoS-class planes, o-QCs) that have been bought to service peers.
- QoS performance characteristics: are a set of QoS parameters like loss rate, one-way packet delay and one-way delay variation.

5.5.1.5.1 QoS Service capabilities

It is useful for a q-BGP peer to know the capabilities of a q-BGP neighbour with respect to the q-BGP protocol extensions and the supported attributes. Capabilities negotiation is achieved thanks to the specification of a new optional parameter that has been specified in [RFC3392]. This parameter is included in the optional parameters of the OPEN message of a q-BGP connection.

In order to indicate that a given inter-domain QoS delivery solution (in the context of MESCAL, we speak about solution options) belongs to a given group (either group-1 or group-2), we introduce a new parameter called *QoS Service Capabilities*. A q-BGP speaker should use this capabilities advertisement in order to indicate the group to which an offered inter-domain QoS delivery solution belongs to, so that its peers can deduce if they can use the 'QoS service'-related attributes with this service peer

The fields of this optional service options capabilities parameter are set as follows:

- The capability code field is set to a value between 128 and 255 as described in [RFC2434]
- The capability length is set to 2
- The capability value field is encoded as shown in Figure 61:

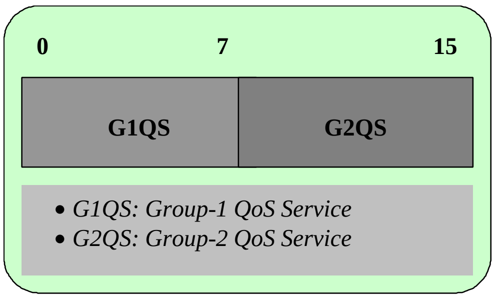


Figure 61. QoS service capability attribute.

- The first octet is set to 0xFF if an offered inter-domain QoS delivery solution that belongs to group-1 is supported (in the context of MESCAL, if a given domain offers the statistical solution option)
- The second octet is set to 0xFF if an offered inter-domain QoS delivery solution that belongs to group-2 is supported (in the context of MESCAL, if a given domain offers the loose and/or the hard solution options)

5.5.1.5.2 QoS Class identifiers

5.5.1.5.2.1 Overview

In order to advertise QoS-related information in q-BGP messages, a dedicated field in q-BGP messages will be introduced. The field is called "*QoS Class identifier*". This field carries the information about the PDB, meta-QoS-class or o-QC (depending on deployed inter-domain QoS delivery solution) that is used in the downstream AS. The value of this field conforms to what has been agreed between two service peers during pSLSs negotiation phase. Note that QC identifiers could be different than the DSCP code point.

The proposed field length is an octet and it is inserted in the QOS_NLRI attribute.

5.5.1.5.2.2 MESCAL Specific requirements on QC identifiers

As mentioned above, the QC identifier' purpose is to indicate to service peers either a meta-QoS-class plane or an o-QC a given q-BGP announcement belongs to. Note that specific range of QC identifiers has to be assigned for meta-QoS-class usage so as to allow global use of the meta-QoS-class concept and then global and unified usage of this concept. If not, the value of the meta-QoS-class identifier will be negotiated and agreed between two service peers.

Within the context of the loose and the hard solution option, 64 possible values of meta-QoS-class identifiers are sufficient since the number of meta-QoS-class identified (at least for the MESCAL project) is less than that. Only 4 or 5 meta-QoS-classes are judged pertinent to standardise. Definition of new meta-QoS-classes is possible since the concept is open and is basically based on the requirements of applications.

For the statistical solution option, and in a large-scale environment, 64 could be easily consumed. This could be a handicap for this solution option. Aggregation methods and pertinent service objectives are to be considered carefully within this solution option.

5.5.1.5.3 QoS-related information

5.5.1.5.3.1 QOS_NLRI attribute

In order to convey QoS-related information, we adopt the [QOSNLRI] proposal that consists at introducing a new optional transitive attribute called QOS_NLRI attribute as the starting point. Some modifications are added to the [QOSNLRI] proposal in order to meet the requirements listed in the sections above. The modifications are twofold:

- Information carried by this attribute:
  - The [QOSNLRI] proposal allows to send only one QoS performance characteristic per q-BGP announcement. This limitation has been relaxed within this specification since it might be necessary to carry a list of QoS performance characteristics in a single q-BGP UPDATE message.
  - Information about QC identifiers: unlike the [QOSNLRI] proposal, this specification allows to propagate information about extended QCs that are pre-negotiated between service peers. Thus PDB, meta-QoS-class and/or o-QC identifiers are announced by q-BGP thanks to QOS_NLRI attribute.
  - The [QOSNLRI] proposal adopts the multiple paths [Walton]. This isn't the case of the current specification in the current stage of the MESCAL project.
  - The PHB identifier has been removed from the list of possible "QoS Information Code" because of the existence of "QoS Class identifier"
- The format of the QoS_NLRI attribute:
  - Add a new field called "QoS Information length": the purpose of this field is the control of the list of QoS performance characteristics that are enclosed in a q-BGP UPDATE message. The use of this field isn't detailed in the current specification. Additional checksum methods could be considered.
  - The lengths of "QoS Information code" and "QoS Information Sub-code" have been reduced to 4 bits in order to reduce the total length of the QOS_NLRI attribute. This is also motivated by the fact that 2^4 values are sufficient to indicate this information.

This attribute is encoded as shown in Figure 62.

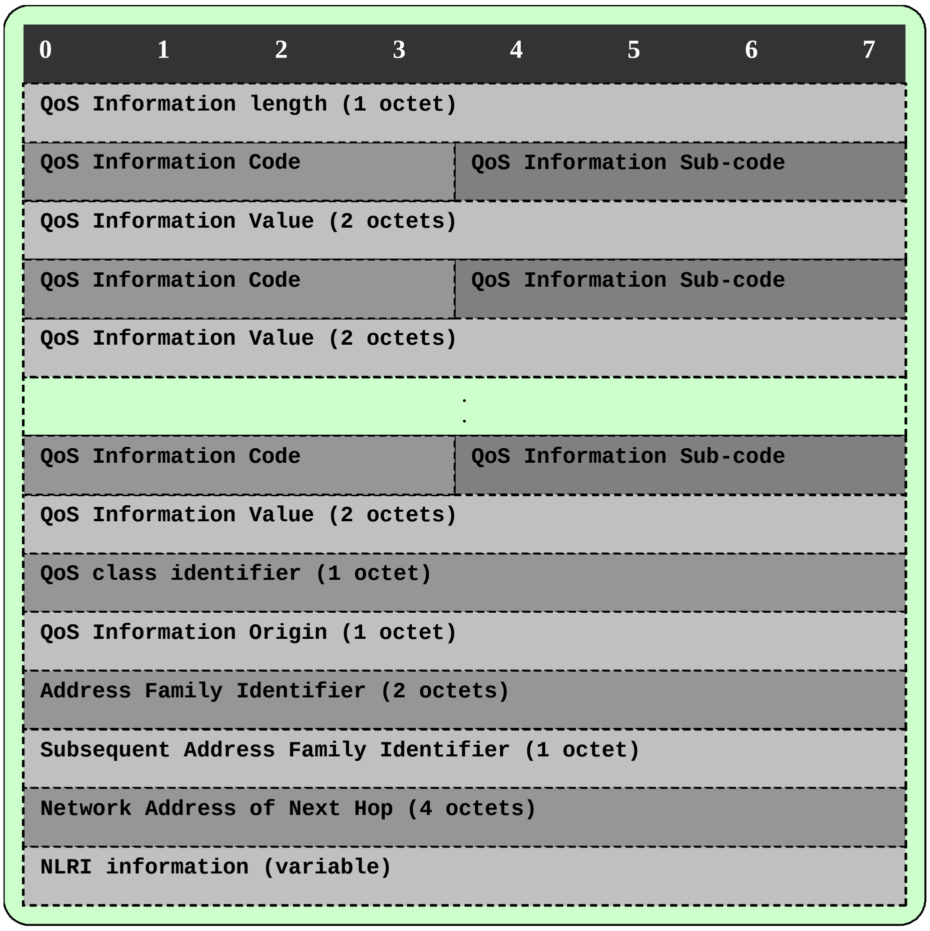

0 1 2 3 4 5 6 7
QoS Information length (1 octet)
QoS Information Code
QoS Information Sub-code
QoS Information Value (2 octets)
QoS Information Code
QoS Information Sub-code
QoS Information Value (2 octets)
QoS Information Code
QoS Information Sub-code
QoS Information Value (2 octets)
QoS class identifier (1 octet)
QoS Information Origin (1 octet)
Address Family Identifier (2 octets)
Subsequent Address Family Identifier (1 octet)
Network Address of Next Hop (4 octets)
NLRI information (variable)

Figure 62. QoS_NLRI attribute.

The meaning of the fields of the QOS_NLRI attribute is defined below:

- QoS information length: this field carries the number of the QoS information Code that will be sent by the BGP speaker in a single q-BGP UPDATE message.
- QoS information Code: this field identifies the type of QoS information:
  - (0) Reserved
  - (1) Packet rate
  - (2) One-way delay metric
  - (3) Inter-packet delay variation
- QoS information Sub-code: this field carries the sub-type of the QoS information. The following sub-types have been identified:
  - (0) None
  - (1) Reserved rate
  - (2) Available rate
  - (3) Loss rate
  - (4) Minimum one-way delay
  - (5) Maximum one-way delay
  - (6) Average one-way delay

Table 11 summarises the compatible (code, sub-code) pairs (a red cell refer to an invalid pair).

	0	*1*	*2*	*3*	*4*	*5*	*6*
0	🟩	🟩	🟩	🟩	🟩	🟩	🟩
1	🟨	🟨	🟨	🟨	🟥	🟥	🟥
2	🟩	🟥	🟥	🟥	🟩	🟩	🟩
3	🟨	🟥	🟥	🟥	🟥	🟥	🟥

Table 11. Compatible (code, sub-code) pairs

- QoS information value: this field indicates the value of the QoS information. The corresponding units depend on the instantiation of the QoS information code.
- QoS information origin: this field provides indication on the origin of the path information.
- QoS class identifier: this field indicates the QC identifier as described in [DS].
- Address Family Identifier (AFI): this field carries the identity of the Network Layer protocol associated with the Network Address that follows.
- Subsequent Address Family Identifier (SAFI): this field provides additional information about the type of the prefix carried in the QOS_NLRI attribute.
- Network address of Next Hop: this field contains the IPv4 Network Address of the next router on the path to the destination prefix.
- Network Layer Reachability Information: This variable length field lists the NLRI information for the feasible routes that are being advertised by this attribute. The next hop information carried in the QOS_NLRI path attribute defines the Network Layer address of the border router that should be used as the next hop to the destinations listed in the QOS_NLRI attribute in the UPDATE message.

5.5.1.5.3.2 MESCAL Considerations

As already mentioned in this document, QoS performance characteristics will be only provided when either the loose or the hard solution option is deployed. In the case of the statistical solution option, no QoS performance characteristic is propagated by q-BGP messages.

As far as the loose and the hard solution options are concerned, the QoS-related information characterises the QoS performance of the route within the meta-QoS-class specified by the value of the attached QoS Class Identifier field.

Within the context of inter-domain QoS delivery solutions that make use the meta-QoS-class concept, a priority property will be associated to each QoS performance characteristic. For example: for loss sensitive meta-QoS-classes a value indicating a high priority could be assigned to loss rate parameter. This usage and knowledge of this priority value is part of the definition of the meta-QoS-class and is suppose to become well known from providers. Therefore, there is no need to propagate these priority properties in q-BGP messages.

5.5.1.5.3.3 Additional features

[Walton] proposes a mechanism that allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous ones. This is achieved thanks to the use of an arbitrary identifier that will identify (in addition to the prefix) a given path. This feature is important but won't be considered in the current stage of the MESCAL project since the amount of reachability information will be huge and could impact the stability and the scalability of q-BGP.

5.5.1.5.4 MESCAL specific change: Service options differentiation

As a result of the study about the inter-working of the MESCAL solution options, it was decided that a clear differentiation had to be introduced in order to separate the q-BGP announcements in order to solve signalling problems (see [D1.4]). This could be implemented by dedicating specific community values for each solution option.

The BGP community attribute was added to BGP in order to simplify the configuration of complex routing policies. It is an optional and non-transitive attribute. The community attribute is a list of community values that are between 0x00000000 and 0xFFFFFFFF. The ranges 0x00000000 through 0x0000FFFF and 0xFFFF0000 through 0xFFFFFFFF are reserved. The well-known values are:

- 0xFFFFFF01: NO-EXPORT. If a BGP speaker receives a route with this community value, the BGP speaker must not export that route beyond its local AS.
- 0xFFFFFF02: NO-ADVERTISE. If a BGP speaker receives a route with this community value, it must not re-advertise this route beyond the local router.
- 0xFFFFFF03: NO-EXPORT-SUBCONFED. This is similar to the NO-EXPORT except that it is used in the context of the confederations.

[RFC1997] states:

" *The rest of the community attribute values shall be encoded using an autonomous system number in the first two octets. The semantics of the final two octets may be defined by the autonomous system (e.g. AS 690 may define research, educational and commercial community values that may be used for policy routing as defined by the operators of that AS using community attribute values 0x02B20000 through 0x02B2FFFF).*"

In order to be aligned with this recommendation and also to allow the distinction between the supported service options, only the second octet of the community value will be used to indicate the service option. The first octet will encode the AS number (NAS). Thus, we adopt the following structure:

- NAS:01- this means that the AS identified by the number NAS supports the loose solution option.
- NAS:02- this means that the AS identified by the number NAS supports the statistical solution option.
- NAS:03- this means that the AS identified by the number NAS supports the hard solution option.

5.5.1.5.5 Route selection process

5.5.1.5.5.1 Classical Route selection process

The BGP specification [RFC1771] has defined a decision process for the selection of the routes that will be installed in the local RIB. This process is responsible for the:

- Selection of routes to advertise to BGP listeners located in the local speaker's autonomous system
- Selection of routes to advertise to BGP listeners located in neighbouring autonomous systems
- Route aggregation and route information reduction.

This process takes into account the BGP attributes, which can impact the selection of the routes. [RFC1771] specifies a set of attributes that could be used as tie-breaker in the context of the route selection process. The following attributes are the most used:

- LOCAL_PREF: this attribute is used at the beginning of the route selection process. It is a well-known discretionary attribute that is used by a BGP speaker to inform BGP peers from its own autonomous system of the originating speaker's degree of preference for an advertised route.
- MED: this attribute is an indicator of which local entrance point an AS would like a peering AS to use. This attribute isn't suitable to break the tie between two equal paths learned from distinct ASs.
- IGP Metric: this metric could be used to influence the choice of the path to put in the local RIB.

Hereafter the BGP Path selection process as commonly understood and implemented:

- Prefer largest Local Preference.
- If same Local Preference prefer the route that the specified router has originated.
- If no route was originated prefer the shorter AS path.
- If all paths are external prefer the lowest origin code (IGP<EGP<INCOMPLETE).
- If origin codes are the same prefer the path with the lowest MED.
- If path is the same length prefer the External path over Internal.
- Prefer the route with the lowest IP address value for BGP router ID.

This process may vary from a vendor to another. For instance, the Cisco implementation adds a new metric called "weight" that is used to choose the best path.

5.5.1.5.5.2 Modified Route selection process

As far as QoS-related information are conveyed in BGP UPDATE messages, the route selection process should take into account this information in order to make a choice and break the tie between equal paths and determine the one(s) to be stored in the local RIB. This process could differ between solutions that belong to group-1 or group-2 (in the MESCAL context between the loose and the statistical solution option).

5.5.1.5.5.2.1 Group-1

In the context of an inter-domain QoS-delivery solution that belongs to group-1 (example: the loose solution option), q-BGP UPDATE messages carry QoS performance characteristics (in the context of the loose solution option, these QoS performance characteristics are the concatenation of all the local QoS class performance characteristics implemented by traversed ASs). This information must be taken into account in order to determine the path that will be stored in the local RIB.

In this case, the route selection process becomes:

```
1. Consider routes that serves a same destination
2. Consider routes that have the same QoS class identifier
3. Compare the QoS performance characteristics associated with
   resulting routes with respect to a well know comparison logic
4. Return the route that optimises the QoS performance
   characteristic
```

In the case of the loose solution option, comparison logic could be based on the use of the priority value that has been affected to each QoS performance characteristic. Therefore, the step 3, could be developed as follows:

```
1. Consider the resulting routes
2. Consider the QoS performance characteristic that has the highest
   priority, and return the routes that optimise that QoS
   performance characteristic
3. If only one route is returned
       Store this route in the RIB
4. If more than one route are returned
       ➔Exclude the QoS performance characteristic that has been used
       in the step 2 from the list of QoS performance characteristic.
       ➔Go to step 1.
```

5.5.1.5.5.2.2 Group-2

As far an inter-domain QoS-delivery solution that belongs to group-2 (example: the statistical solution option) is considered, only the identifier of the QC (o-QC in the context of the statistical solution option) has to be taken into account in order to choose a path that will be stored in the local RIB. The modified route selection process will be as follows:

```
1. Consider the received routes that served the same destination
2. Consider the routes with similar o-QC identifier
3. Apply local policies (prefer a given origin AS, cost,…).
4. If only one route has been returned
       Store this route in the RIB
5. If more than one route has been returned
       Apply the classical BGP route selection process.
```

5.5.1.5.6 Additional features

Other concepts need to be studied, such like MC-aggregation in order to reduce the volume of exchanged information.

5.5.1.5.6.1 Test objectives

The q-BGP protocol is used as a technical means to support the MESCAL solution options and must provide the element of service that are necessary for the MESCAL solution options to operate correctly. At a high level of description, q-BGP provides two main basic functions, which are: the discovery of inter-domain QoS routes and the selection and enforcement of an optimal QoS path. The protocol itself embeds behavioural and related inter-processes communication aspects that must be evaluated and tested.

The objectives of the tests are to verify that the q-BGP specification provides MESCAL with the requested functionality. In particular testing specification and activities shall verify that the protocol:

- Provides inter-domain DSCP-aware routing functions.
- Provide route selection functions per DSCP-plane.
- Is able to optimise the end-to-end QoS performance characteristics of a route using QoS information exchanged between the domains.
- Can take into account and can apply directives from offline TE related blocks.
- QOS_NLRI attribute definition is suitable for carrying efficiently all relevant QoS related information.

Additional testing activities will be achieved to validate the pertinence of the service and solution options specified by the MESCAL project. Those latter tests will be specified in deliverable D3.1.

5.5.2 PCS Communication Protocol

5.5.2.1 Introduction

MESCAL has specified three solution options that target distinct categories of customer and that offer different services guarantees. This section focuses on the hard solution option (i.e. solution option 3), which has been designed to offer strict QoS guarantees for the corporate market (hard service option in conformance with MESCAL terminology).

This section presents the Path Computation Service, its interactions with the MESCAL functional blocks and provides a description of a first version of the PCS Communication Protocol. This specification has been primarily realised to serve proof of concept considerations. It will be implemented, tested and evaluated during the next phases of the project. An updated version will be produced and integrated in [D1.3]

The following discussions rely on the inter-domain QoS signalling solution described in section 5.6.2.5.2 from D1.4.

5.5.2.2 PCS review

The hard solution option makes use of a particular entity called PCS (Path Computation Server), which is responsible for finding an inter-domain path satisfying a set of QoS performance guarantees to establish inter-domain QoS LSPs. The computation of this path is distributed and needs PCSs from different domains to communicate. The communication between two PCS entities is enabled by the PCS Communication Protocol (PCP). Once "computed", the path is provided to the RSVP-TE/MPLS machinery of the head-end LSR, which can establish an inter-domain LSP that will follow the inter-domain path provided by the PCS.

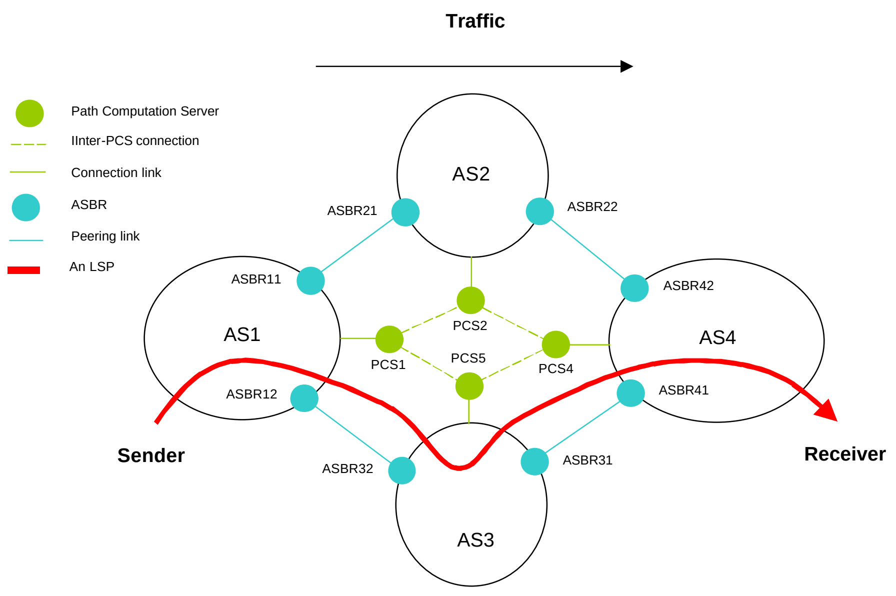


Figure 63. Overview

In Figure 63 above, each domain is assumed to support a set of meta-QoS-class implemented by l-QCs and hard service option pSLSs have been established between the domains. q-BGP is consequently running between the domains and each domain learns, per meta-QoS-class plane, the set of PCSIDs of the remote domains that can be reached, together with some aggregated QoS performance characteristics. The PCSID learnt by the peering ASBR can be distributed to the other routers of the domain, via the q-iBGP mesh (or the use of route reflectors).

A PCS is present in each domain that supports the hard service option. This PCS also receives q-iBGP announcements from all the ASBRs of its domain. Thus, the PCS knows per *meta-QoS-class* plane, all the remote ASs (via the PCSID) that support the hard solution option together with the associated QoS performance guarantees associated with their inter-AS path.

Each time a hard service option pSLS is established, the domains exchange their respective PCS information (name, IP address, identifiers, authentication information…) so that they can communicate.

In order to create an inter-domain QoS LSP, the domain which requests the establishment of the LSP asks its PCS to compute an inter-domain path satisfying QoS constraints, expressed in term of *Meta-QoS-Class* availability along the path together with optional additional constraints such as: an associated bandwidth guarantee per *Meta-QoS-Class* and/or a maximum end-to-end delay for instance. The first PCS selects one possible path among the set of alternatives and identifies the next-hop domain. It then verifies that the appropriate resources are available in its own domain and sets up administrative pre-reservations in the management system of its domain. Then it contacts the next hop PCS, requesting a path computation between the next hop ASBR and the termination address of the inter-domain LSP. This second PCS performs the same computation as the first one and the procedure is iteratively repeated up to the last PCS. If a path satisfying all requirements is found, each PCS returns the path received from the responding PCS concatenated with the sub-path it computed. When the last result reaches the originating PCS the whole path is available.

5.5.2.3 PCS service

From the hard service option management point of view, the key service requested to the PCS is mainly to provide a path computation service, which consists in finding an inter-domain path satisfying a set of QoS constraints. Other services or variants from this key service could be imagined and requested by the hard service option management such as: requesting a path computation for informational purposes or cancelling a request in progress. These advanced features are not hereafter considered and are left for further study.

5.5.2.3.1 Overview

The deployment and the maintenance of the hard service option require the cooperation of several functional blocks from the MESCAL functional model. Within the MESCAL functional mode, PCS is "only" responsible for computing an inter-domain QoS path. The implementation of the service (whether it is automated or not) and the creation of inter-domain LSP results from the cooperation of functional blocks, including management plane blocks, control plane blocks and data plane blocks.

The PCS does not itself trigger the establishment of any inter-domain LSP, but provides inter-domain paths, when those are available. In particular, it is un-aware of business considerations but the hard service option management is. The PCS provides an interface for the higher-level functional blocks so that they can ask for path computation when necessary. It communicates with other remote PCS thanks to the PCP protocol and requests additional services from other functional blocks as illustrated in Figure 64.

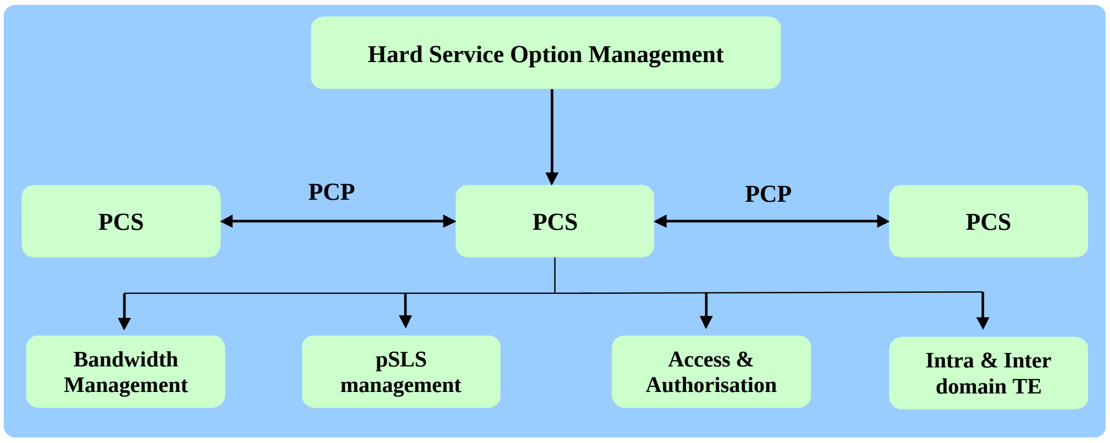


Figure 64. PCS interfaces

5.5.2.3.2 Interactions with MESCAL functional blocks

A pSLS established within the scope of the hard service option formalises a right to establish inter-domain LSPs. But the destination and the number of future LSPs is not known in advance. The pSLS indicates only the upper-boundaries that the upstream AS is allowed to use, in terms of meta-QoS-class that can be used at the inter-domain for establishing inter-domain LSP and in terms of maximum bandwidth associated to each meta-QoS-class. The pSLS does not reserve any network resources in advance and the cost associated to such pSLS should be relatively small. On the other hand, resources are actually allocated when an inter-domain LSP is set up and the costs associated with the service depend on the characteristics of the LSP. The management plane of each downstream domain along the path should be aware of the existence of those LSPs together with their associated QoS guarantees in order to provide an accurate bill to the upstream AS.

However, it is difficult to establish such a contract in advance especially when the LSP path is not known. Thus, the sequence of operation for establishing a LSP should be:

- Compute an inter-domain path.
- Negotiate inter-domain contracts along the path for this particular LSP using information returned by the path computation.
- Establish the LSP once final contractual terms have been end-to-end agreed.

The establishment of this cascade of contracts can be difficult to achieve and can take some time. In particular, the risk is not negligible that the resources that were available when the PCS performed the path computation are no longer available along the path when the cascaded contracts are agreed, because others LSPs have used the corresponding resources.

In order to solve this issue it is necessary that the PCS of each domain makes an administrative reservation of the corresponding resources and indicates the characteristics of the path. This information is registered by the management plane which triggers in parallel the creation of a provisional contract referencing the technical characteristics of the future LSP. Resources are now administratively reserved. Subsequent path computation requests may be impacted because the management plane removes these resources from the available overall network resources. This provisional contract is valid for a limited time, which is the minimum date reported by each domain along the path. If the date exceeds this limit the provisional contract (or pre-contract) can be removed from the management systems, and related administrative network resources have to be relaxed.

It is the responsibility of the management plane of each domain to cooperate in agreeing the exact financial terms and additional clauses of this contract, including its duration. Each domain knows the entry and the exit point of the LSP within its own domain and consequently knows both the upstream and downstream ASs to deal with. This validation procedure should ideally be automated to speed up the process and could integrate pricing negotiation. The way that the other blocks of the management plane deal with this automation if out the scope of this study.

Thus, once the pre-contract is validated, the path computed by the PCS can be provided to the head-end LSP, which effectively sets up the LSP. Note that each ingress point of each domain should activate some outsourced policy functions that would allow RSVP TE to get an agreement from the management platform.

The PCS interacts also with the intra and inter-domain TE blocks to retrieve routing information that is used to compute an inter-domain path satisfying expressed QoS constraints. An interface must be made available to the PCS so that it can access this information. Note that both intra and inter-domain routes must be made available to the PCS

In addition, for access control and authorisation purposes, the PCS must be provided with access to the list of other PCSs from which it will accept requests. This list is updated each time a hard service option pSLS is agreed by the provider.

5.5.2.3.2.1 Finding an inter-domain path

In order to find an inter-domain path, the PCS must be provided with the head-end and tail-end characteristics of the LSP terminations. Each termination description must include the loopback IP address of the LSP end-point and the PCSID of the domain owning the corresponding resources. This description will be represented in the form: IPAddress@PCSID in the rest of this section. This information must also include the QoS performance guarantees required for the inter-domain QoS LSP. This information encompasses the requested meta-QoS-classes so that the set of collaborating PCS can compute a path that will cross a set of domain supporting the requested meta-QoS-classes. It can also contain, per meta-QoS-class, additional QoS performance guarantees the PCS must take into account. These are for example the guaranteed end-to-end delay, jitter, loss rate or bandwidth. Note that these parameters can differ depending on the meta-QoS-class and they may not all be present in the request. If present they must be taken into account by the PCS. If the PCS doesn't understand the QoS parameter, the PCS must stop its computation and must return an appropriate error.

When computing a path, a PCS interacts with other blocks from the management plane. In particular it checks the availability of the resources within and at the boundaries of the domain. If the resources are available and the sub-path (path between the ingress point of the domain and a potential ingress point of a peering domain) conforms to the path constraints requested it must inform the management plane of a pre-reservation concerning this path so that other path computation requests can take this information into account. Once achieved, the request if propagated to the PCS of the next domain which has been selected by the PCS.

The request is not propagated as is. In particular the QoS performance guarantees must be updated to reflect the QoS performance guarantees already experienced along the path. In the case of end-to-end delay, the end-to-end delay must be computed in such a way it reflects the delay requested between the ingress point of the next domain up the tail-end termination of the LSP. In order to achieve this computation the PCS must be aware of the QoS performance guarantees of the meta-QoS-classes that its domain supports.

A globally well-known meta-QoS-class identifier can be used between the PCSs, but if the providers did not agree to use it, the requesting PCS must use the identifier they agreed to use during the pSLS negotiation phase.

In some particular cases, mainly dictated by business relationships constraints, the initial requestor may wish to exclude particular ASs from the path computation. In such a case, additional constraints must be added. These constraints can be expressed using the AS number of the domains.

If an end-to-end LSP has to be re-engineered because the associated constraints have changed in terms of meta-QoS-class requested, bandwidth, delay… a new end-to-end path needs to be computed. In order to improve its chances of finding a valid path, the requestor can specify that the path for which the request is issued will replace a previously established LSP. For doing so, the requestor can indicate the reference of the path corresponding to this LSP. A PCS can release, during the path computation process, the resources corresponding to the former LSP, if the new path follows part of the former path. This reference is stored in the management plane of each domain and is generated by the initial requestor. This reference is globally unique.

The ability to indicate such additional constraints can be interesting in the case of backup LSPs so that the PCS can compute a path using distinct resources. These considerations are for further study.

5.5.2.4 The PCS Communication Protocol (PCP)

5.5.2.4.1 Overview

This Section describes a simple query and response protocol that can be used between PCS entities to collaborate for computing an inter-domain QoS constrained path.

The main characteristics of the PCP protocol include:

- The protocol employs a client/server model in which a PCS can both act as a client and/or a server at the same time. A client PCS sends requests, cancellation and receives responses.
- The protocol uses TCP as its transport protocol for reliable exchange of messages between PCS. Therefore, no additional mechanisms are necessary for reliable communication between two PCS.
- In this first version, PCP does not provide any message level security for authentication, replay protection, and message integrity. But PCP can reuse existing protocols for security such as IPSEC or TLS to authenticate and secure the channel between two PCS.
- The PCP protocol described below supports only a basic path computation service. In particular it doesn't support additional path computation constraints, nor enhanced reporting features in case of path computation failure.

5.5.2.4.2 PCP messages

This section discusses the PCP message formats and objects exchanged between PCS.

5.5.2.4.2.1 Common header

Each PCP message consists of the PCP header followed by a number of arguments depending on the nature of the operation.

```
         0            1            2            3
  +------------+------------+------------+------------+
  |  Version   |  Op Code   |       Message Length    |
  +------------+------------+------------+------------+
```

Global note: //// implies field is reserved, set to 0.

The fields in the header are:

Version: 8 bits. PCP version number. Current version is 1.

Op Code: 8 bits. The PCP operations are:

1 = OPEN	(OPN)
2 = ACCEPT	(ACP)
3 = CLOSE	(CLO)
4 = REQUEST	(REQ)
5 = RESPONSE	(RSP)
6 = PATH-ERROR	(ERR)
7 = CANCEL	(CCL)
8 = ACKNOWLEDGE	(ACK)
9 = KEEP-ALIVE	(KA)

Message Length: 16 bits

This is the size of the message in octets, which includes the standard PCP header and all encapsulated objects. Messages MUST be aligned on 4 octet intervals.

5.5.2.4.2.2 OPEN message

```
       0            1             2            3
+------------+------------+------------+------------+
|                                                   |
|                      PCSID                        |
|                                                   |
|                                                   |
+------------+------------+------------+------------+
```

The message contains only one argument. This PCSID is propagated by q-BGP between the domains. This is a routable IPv4 or IPv6 address identifying a PCS of a domain. This PCSID must be inserted by the PCS opening a PCP session. The size of the PCSID is 4 or 16 bytes.

5.5.2.4.2.3 ACCEPT message

```
       0            1             2            3
+------------+------------+------------+------------+
|        KA-Timer         |///////////////////////////|
+------------+------------+------------+------------+
```

- Keepalive (KA)-Timer: The argument of the accept message is a 2 octets integer value which represents a timer value expressed in units of seconds. This timer value is treated as a delta. KA-Timer is used to specify the maximum time interval over which a PCP message MUST be sent by the two communication entities. The range of finite timeouts is 1 to 65535 seconds represented as an unsigned two-octet integer. The value of zero implies infinity.

5.5.2.4.2.4 CLOSE message

The close message contains an error code indicating the reason of the close of the session.

```
        0            1             2            3
+-------------+-------------+-------------+-------------+
|          Error-Code       | ////////////////////////////|
+-------------+-------------+-------------+-------------+
```

- Error-Code:

  1 = Shutting Down

  2 = Bad Message Format

3 = Incorrect identifier

4 = Unable to process

5 = Protocol error

5.5.2.4.2.5 REQUEST message

The Request message is sent by the client PCS for computing and inter-domain path.

```
       +-------------+
1 byte |    TTL      |
       +-------------+
1 byte |     L0      |
       +-------------+-------------+-------------+-------------+
2 bytes|                       AS-NUMBER                       |
       +-------------+-------------+-------------+-------------+
       //                                                      //
       +-------------+-------------+-------------+-------------+
2 bytes|                       AS-NUMBER                       |
       +-------------+-------------+
2 bytes|            L1             |
       +-------------+-------------+-------//----+-------------+
       |                  PATH-COMPUTATION-ID                  |
       |-----------------------------------//------------------|
2 bytes|            L2             |
       +-------------+-------------+-------//----+-------------+
       |                  PATH-REFERENCE-ID                    |
       +-------------+-------------+-------//----+-------------+
2 bytes|                  REQ-REFERENCE-ID                     |
       +-------------+-------------+-------------+-------------+
1 byte |   ADD-TYPE  |
       +-------------+-------------+-------//----+-------------+
       |                  HEAD-END-ADDRESS                     |
       +-------------+-------------+-------//----+-------------+
       |                  TAIL-END-ADDRESS                     |
       +-------------+-------------+-------//----+-------------+
1 byte |  NUMBER-OF-QC-CONSTRAINT  +
       +-------------+-------------+
2 bytes|   QC-CONSTRAINT-LENGTH    +
       +-------------+-------------+
1 byte |   QOS-CLASS-IDENTIFIER    +
       +-------------+-------------+---------------------------+
1 byte |   QOS-INFO-CODE           +   QOS-INFO-SUB-CODE       |
       +-------------+-------------+-------------+-------------+
2 bytes|                  QOS-INFO-VALUE                       |
       +-------------+-------------+-------------+-------------+
       |   QOS-INFO-CODE           +   QOS-INFO-SUB-CODE       |
       +-------------+-------------+-------------+-------------+
       |                  QOS-INFO-VALUE                       |
       +-------------+-------------+-------------+-------------+
       |   QOS-INFO-CODE           +   QOS-INFO-SUB-CODE       |
       +-------------+-------------+-------------+-------------+
       |                  QOS-INFO-VALUE                       |
       +-------------+-------------+-------------+-------------+
```

- TTL: is the maximum number of ASs that can be crossed by the path. This field is decremented by one each time a PCS issues a request.
- L0: is a 1-byte length field. It represents the number of ASs that have already been crossed.
- AS-NUMBER: is a 2 bytes length field representing an AS number. The first AS-NUMBER value of the list is the AS-NUMBER of the initial PCS a path computation.

- L1: is the length in bytes of the PATH-COMPUTATION-ID. Size of this field is 2 bytes.
- PATH-COMPUTATION-ID: is a globally unique value that identifies a path computation occurrence. It is a variable-length field. It is suggested, at least in this first specification, that this identifier is computed using the PCSID of the domain, concatenated with the date and finally an identifier that will be computed by the first requesting PCS each time a request will have to be issued. Across PCS reboots, this identifier must be unique. This PATH-COMPUTATION-ID will be replicated in all subsequent request initiated by the PCSs along the path.
- L2: is the length in bytes of the PATH-REFERENCE-ID. Size of this field is 2 bytes.
- PATH-REFERENCE-ID: is a variable-length field. It is an identifier that represent a pre-agreement between the head and the tail-end domain that allows the PCS from the terminating domain to accept or reject the path computation request.
- REQ-REFERENCE-ID: is a 2 bytes length field representing an unsigned integer. This field is used to identify the REQUEST. It allows making the difference between several REQ issued for different path computation (but same PATH-COMPUTATION-ID) between two neighbour ASs interconnected via multiple links.
- ADD-TYPE: indicates the nature of the IP addresses of the tail-end and head-end termination:

  1 = IPv4

  2 = IPv6
- HEAD-END-ADDRESS: is the head-end address of the future LSP represented in the form HEAD-END@PCSID. This is a couple of IPv4 or IPv6 address. The first address of the couple identifies a loopback or an interface address of a network element, the second element is the PCSID of the domain owning the previous address.
- TAIL-END-ADDRESS: is the tail-end address of the future LSP represented in the form TAIL-END@PCSID. This is a couple of IPv4 or IPv6 address. The first address of the couple identifies a loopback or an interface address of a network element, the second element is the PCSID of the domain owning the previous address.

This above parameters MUST be present in each REQUEST and in the same order.

- NUMBER-OF-QC-CONSTRAINT: represents the number of QoS class constraints the PCS must take into account when computing a path. A QoS class constraint contains a QoS-Class-Identifier (which identifies a particular meta-QoS-class) followed by additional constraints. The size of this filed is 1 byte. This field is not really necessary in this first version of the specification but it could become useful if additional path constraints were included in the request.
- QC-CONSTRAINT-LENGTH: is the length in byte of the QoS-Class-Constraint that follows. The size of this field is 2 bytes.
- QOS-CLASS-IDENTIFIER: identifies a particular meta-QoS-class. The size of the field is 1 byte.
- QOS-INFO-CODE: this field identifies the type of QoS information. The size of this field is 4 bits.
  - (0) Reserved
  - (1) Packet rate
  - (2) One-way delay metric
  - (3) Inter-packet delay variation

QOS-INFO-SUB-CODE: this field carries the sub-type of the QoS information. The following sub-types have been identified. The size of this field is 4 bits.

- (0) None
- (1) Reserved rate
- (2) Available rate
- (3) Loss rate
- (4) Minimum one-way delay
- (5) Maximum one-way delay
- (6) Average one-way delay

- QOS-INFO-VALUE: this field indicates the value of the QoS information. This is the constraints that the PCS should respect. The corresponding units depend on the instantiation of the QoS information code.

5.5.2.4.2.6 RESPONSE-PATH message

This message is sent back when a path has been successfully computed.

```
       +-------------+------------+
2 bytes|            L1            |
       +-------------+------------+-------//----+-------------+
       |                 PATH-COMPUTATION-ID                   |
       |--------------------------------//------------------|
2 bytes|                   REQ-REFERENCE-ID                    |
       |--------------------------------//------------------|
1 bytes| PATH-LENGTH |
       +-------------+
1 byte |   ADD-TYPE  |
       +-------------+------------+-------//----+-------------+
       |                      NEXT-HOP                         |
       +-------------+------------+-------//----+-------------+
       //                                                     //
       +-------------+------------+-------//----+-------------+
       |                      NEXT-HOP                         |
       +-------------+------------+-------//----+-------------+
8 bytes|       VALIDITY-DATE      +
       +-------------+------------+
1 byte |  NUMBER-OF-QC-CONSTRAINT +
       +-------------+------------+
2 bytes|   QC-CONSTRAINT-LENGTH   +
       +-------------+------------+
1 byte |   QOS-CLASS-IDENTIFIER   +
       +-------------+------------+----------------------------+
1 byte |   QOS-INFO-CODE          +   QOS-INFO-SUB-CODE        |
       +-------------+------------+-------------+--------------+
2 bytes|                 QOS-INFO-VALUE                        |
       +-------------+------------+-------------+--------------+
       |   QOS-INFO-CODE          +   QOS-INFO-SUB-CODE        |
       +-------------+------------+-------------+--------------+
       |                 QOS-INFO-VALUE                        |
       +-------------+------------+-------------+--------------+
       |   QOS-INFO-CODE          +   QOS-INFO-SUB-CODE        |
       +-------------+------------+-------------+--------------+
       |                 QOS-INFO-VALUE                        |
       +-------------+------------+-------------+--------------+
```

- L1: is the length in bytes of the PATH-COMPUTATION-ID. Size of this field is 2 bytes.

- PATH-COMPUTATION-ID: is a globally unique value that identifies a path computation occurrence. It is a variable-length field. The value of this identifier must be the same as the one provided in the REQUEST.
- REQ-REFERENCE-ID: is a 2 bytes length field representing an unsigned integer. This field is used to reference the initial REQUEST.
- PATH-LENGTH: indicate the number of next hops that form the path. The size of this filed is 1 byte.
- ADD-TYPE: indicates the nature of the IP addresses in the PATH. The size of this filed is 1 byte.

  1 = IPv4

  2 = IPv6
- NEXT-HOP: IP address of a next hop that is part of the computed path. Size of this field depends on the nature of the IP address.
- VALIDITY-DATE: represents the GMT date after which the computed path returned will not be valid. The size of this field is 8bytes.

These above parameters MUST be present in each RESPONSE and in the same order.

The other parameters have the same meaning than for the REQUEST except:

- QOS-INFO-VALUE: represents the QoS guarantees of the path, for this particular QoS-INFO-CODE parameter (delay, jitter,…) between the ingress ASBR of the responding PCS and the tail-end of the path.

5.5.2.4.2.7 PATH-ERROR message

This message is sent back when a path could not be computed.

```
       +-------------+------------+
2 bytes|            L1            |
       +-------------+------------+-------//----+------------+
       |                  PATH-COMPUTATION-ID                |
       |-----------------------------------//----------------|
2 bytes|                    REQ-REFERENCE-ID                 |
       |-----------------------------------//----------------|
1 bytes|        REASON-CODE       |
       +-------------+------------+
```

- L1: is the length in bytes of the PATH-COMPUTATION-ID. Size of this field is 2 bytes.
- PATH-COMPUTATION-ID: is a globally unique value that identifies a path computation occurrence. It is a variable-length field. This identifier must be the same as the one provided by the REQUEST.
- REASON-CODE: indicate the reason of the failure. Identified failure are:

  1 = No resource available

  2 = Path reference error

  3 = Abnormal termination

  4 = PATH-COMPUTATION-ID already used

  5 = TTL expired

  6 = Loop detected

  7 = Request already handled

5.5.2.4.2.8 CANCEL message

This message is sent by a client or a server PCS when a path computation must be cancelled.

```
       +-------------+------------+
2 bytes|            L1            |
       +-------------+------------+-------//----+------------+
       |                 PATH-COMPUTATION-ID                 |
       |-----------------------------------//-----------------|
2 bytes|                   REQ-REFERENCE-ID                  |
       |-----------------------------------------------------|
```

- L1: is the length in bytes of the PATH-COMPUTATION-ID. Size of this field is 2 bytes.
- PATH-COMPUTATION-ID: is a globally unique value that identifies a path computation occurrence. It is a variable-length field. This identifier must be the same as the one provided by the REQUEST.
- REQ-REFERENCE-ID: is a 2 bytes length field representing an unsigned integer. This field is used to reference the initial REQUEST.

5.5.2.4.2.9 ACKNOWLEDGE message

This message is sent by a client PCS to a server PCS to confirm the reservation of the path. This feature is particularly used when a client PCS launches multiple REQUEST during its path computation phase.

```
       +-------------+------------+
2 bytes|            L1            |
       +-------------+------------+-------//----+------------+
       |                 PATH-COMPUTATION-ID                 |
       |-----------------------------------//-----------------|
2 bytes|                   REQ-REFERENCE-ID                  |
       |-----------------------------------------------------|
```

- L1: is the length in bytes of the PATH-COMPUTATION-ID. Size of this field is 2 bytes.
- PATH-COMPUTATION-ID: is a globally unique value that identifies a path computation occurrence. It is a variable-length field. This identifier must be the same as the one provided by the REQUEST.
- REQ-REFERENCE-ID: is a 2 bytes length field representing an unsigned integer. This field is used to reference the initial REQUEST.

5.5.2.4.2.10KEEPALIVE message (KA)

Message exchanged between to PCS to signal their correct behaviour when no other messages are exchanged.

This message has no argument.

5.5.2.5 Exchange of PCP messages

5.5.2.5.1 Communication

The PCP protocol uses a single persistent TCP connection between a client PCS and a remote Server PCS. One PCS server implementation per server MUST listen on a well-known TCP port number (to be defined). The client PCS is responsible for initiating the TCP connection to the server PCS. The location of the remote PCS is deduced and retrieved from the management plane blocks during the path computation process or at PCS boot via the pSLS management block. PCS can have crossed communication; some are acting as a client role, others as a server role.

5.5.2.5.2 OPEN (OPN)

An OPN message must be sent before any other exchange. As part of the open message, the client PCS provide its PCSID which allows the server to identify the client. It can also use this information to retrieve the client context near its management plane. Only one OPN message can be issued at a time.

If the server PCS receives malformed message it must close the session using the appropriate error code.

5.5.2.5.3 ACCEPT (ACP)

The ACP message is used to positively respond to the OPN message from the client PCS. This message will return to the PCS a timer value object indicating the maximum time interval between keep-alive messages.

If the server PCS refuses the client, it will instead issue a CLOSE message.

The KA-Timer corresponds to maximum acceptable intermediate time between the generation of messages by the PCSs. The timer value is determined by the server PCS and is specified in seconds.

5.5.2.5.4 CLOSE (CLO)

The CLOSE message can be issued by either the client or the server PCS to notify the other that it is no longer available.

The Error code is included to describe the reason for the close.

When issuing a CLOSE both the client and the server must delete all the internal state related to this PCP session. Additionally, all pending requests must be explicitly cancelled using a CCL message in order to free as much as possible all pending resource reservations and/or pre-contracts that could have been established

5.5.2.5.5 REQUEST (REQ)

A request is issued by a client PCS when it has found a potential path toward the target final destination. This request can be issued as a consequence of a request received from another domain it has agreement with or from its own service management plane.

When the service request comes from a remote PCS, the server performs the following tasks:

- (0) If the receiving TTL is zero the PCS must discard the request. The receiving PCS, decrements by one the received TTL value. If the TTL is equal to zero, the request is rejected if the PCS is not the last PCS in the chain. In addition the PCS examines the AS-PATH included in the received REQ and reject it if it finds its own AS number in the list. This mechanism allows avoiding possible loops when a limited set of QoS constraints are provided in the request.
- (1) It checks if the PATH-COMPUTATION-ID of the received REQ is already associated to a pre-contract or contract. If this is the case, it returns a PATH-ERROR message with a reason-code = 4. It checks if the PATH-COMPUTATION-ID and the REQ-REFERENCE-ID of the received REQ are already associated to a pre-reservation record. If a pre-reservation is found, it returns a PATH-ERROR message with a reason-code = 4.
- (2) It considers the HEAD-END-ADDRESS and the TAIL-END-ADDRESS parameters present in the request. The HEAD-END-ADDRESS must indicate a valid entry point in its domain. If not, the PCS returns a PATH-ERROR with an appropriate reason value.
- (3) Then it extracts the PCSID from the TAIL-END-ADDRESS and parses the QoS constraints provided at part of the request message. It has thus identified all meta-QoS-class required together with their associated QoS constraints.
- (4) The PCS achieves some policing and verifies that the request constraints will not exceed the resources negotiated in the pSLS. If resources are exceeded, the PCS returns a PATH-ERROR message.

- (5) If the PCS recognises its own PCSID in the TAIL-END-ADDRESS, it considers the PATH-REFERENCE-ID otherwise it jumps to step (6). If this identifier is known from its management plane, the request is accepted and processing continues on (51). Otherwise the PCS returns a PATH-ERROR message with a reason-code = 2.
    - (51) The PCS computes an intra-domain path and verifies the availability of the resources along this internal path. If available, the PCS interacts with its management plane and create a contract which triggers the administrative reservation of the resources. When interacting with the management blocks, the PCS must provide all information necessary to identify the sub-path it selected. In particular it must provide the PATH-COMPUTATION-ID, the ingress point ASBR address used in its domain and the termination point in its domain. The PCS sends a RESPONSE-PATH message back to the requesting PCS. If resources are not available a PATH-ERROR message is generated.
- (6) It then queries the dynamic inter-domain traffic engineering block with the retrieved PCSID and the list of requested meta-QoS-classes. The dynamic inter-domain TE block returns the available q-BGP announcements. The PCS then verifies whether it can find a next-hop ASBR which announces the PCSID within the requested meta-QoS-class. If it can not find it the procedure stops and a PATH-ERROR message is returned back to the requesting entity with an appropriate reason-code value.
- (7) If one or several next-hops are found, the PCS examines the QoS performance guarantees of the announcements and compare the values with those requested in the request. If it doesn't understand one of the requested QoS constraints, PATH-ERROR message is sent back. Otherwise, QoS constraints are successively compared to those received from q-BGP. All next-hops propagating the set of announcements satisfying the required QoS constraints are kept. The others are left on side.
- (8) For each possible next hop ASBR the PCS checks is there are enough available resources available at the domain boundaries. In particular if some bandwidth guarantees are required the PCS checks if the administrative maximum bandwidth agreed during the pSLS negotiation phase will not be exceeded. If resources are not available the ASBR is left on side and the next ASBR in the list is considered. If resources are available, the PCS pre-reserves the corresponding resources near the management plane. At this stage, the management plane doesn't create any contract since we are not sure that an end-to-end path exists. This pre-reservation can be taken into account by the PCS for subsequent requests. It can use it as a lock and delay the incoming requests or introduce the pre-reservations in its resource availability computation according to the local policy enforced. When interacting with the management blocks, the PCS must provide all information necessary to identify the sub-path it selected. In particular it must provide the PATH-COMPUTATION-ID, the ingress point address of its domain and the ingress point address of the next domain. This latter information can be used by the management plane to identify the upstream and downstream involved domains.
    - (81) The PCS computes an intra-domain path and verifies the availability of the resources along this internal path. If resources are available, the sub-path is valid and the PCS forms a new REQUEST message which is sent to the PCS of the remote domain owning the next-hop ASBR. It adds its own AS number to the existing list. If internal resources are not available, the PCS discard the pre-reservation and considers the next hop ASBR in the list. When building the request the client PCS keeps the PATH-COMPUTATION-ID, the PATH-REFERENCE-ID, the TAIL-END-ADDRESS unchanged. The initial HEAD-END-ADDRESS is replaced by the address of the ingress next-hop ASBR identified during the path computation. The QoS constraints characteristics are modified in order to take into account the QoS performance guarantees provided by the domain. If for instance a constraint requires a 100 ms delay and the delay guaranteed by the local QoS class of the AS is 15 ms, the value of the corresponding constraint will take the value 85 ms in the new request.
- (9) If QoS constraints cannot be satisfied for any of the ASBR, the PCS returns a PATH-ERROR message.

Note that it is quite possible that several next hops ASBR can satisfy the requested constraints. In such a case the PCS can process one next-hop ASBR at a time or several in parallel. For one incoming request, there can be multiple simultaneous outgoing requests towards different PCS. If several requests are sent toward the same neighbour, for a same PATH-COMPUTATION-ID, the REQ-REFERENCE-ID must be different. Nevertheless, this feature can lead to scalability issues and needs further investigations.

5.5.2.5.6 RESPONSE (RSP)

A RESPONSE message is sent by a PCS server in response to a request issued by a client PCS. RSP messages are sent back when a valid end-to-end path has been computed. The RSP message is necessarily initiated by the tail-end domain.

When a valid end-to-end path has been computed, the PCS of the last domain on the path, forms a RSP message. It first inserts the original PATH_COMPUTATION-ID. Then its forms a path argument that must contains the IP address of the tail-end LSP and the IP address interface of the ingress ASBR supporting that path. It may insert between these two extremities, the IP address of additional hops. It also indicates the date after which the path will not be valid anymore because administratively reserved resources will have been relaxed. Then, it indicates the QoS guarantees it really provides between the ingress ASBR and the tail-end address of the LSP. The RSP message is then sent to the requesting PCS.

On receipt, the PCS adds its own intra-domain sub-path to the list. It does not indicate the next-hop ASBR since this has already been inserted by the downstream PCS. This sub-path can be a strict or loose description. It also modifies the QoS guarantee parameters so that they reflect the QoS guarantees it can provide for its part of the path. This is achieved in the same way than for the request, but it is an "addition" operation if we consider the delay, for example. The VALIDITY-DATE is modified so that the value indicates now the smaller date between the date received in the RSP message and the date reported by the management plane.

If the PCS sent multiple REQUEST message in parallel, it must wait for a RSP or ERR message for all the requests it sent. If the PCS got multiple RSP messages it must select only one and inform the un-selected PCS that they can cancel their reservation. It forms CANCEL messages, sends them to the appropriate PCS and cancels its own pre-reservation for the corresponding requests. If the PCS doesn't wish to wait for a reply, it can send a CANCEL message at any time.

The PCS can send the consolidated RES message to the requesting PCS after sending ACK message to the PCS it decided to keep in the path.

5.5.2.5.7 ACKNOWLEDGE (ACK)

On receipt, the ACK message is used by the PCS server to confirm to its management plane that the resources needed for the path referenced by PATH-COMPUTATION-ID present in the message need to be reserved. In particular it allows the management plane to create a contract based on information previously stored by the PCS during the computation phase. If no ACK is received, no contract is created and the negotiation at the management level will fail. If for some reasons, no ACK were received, the VALIDITY-DATE would be used and the administrative pre-reservation automatically removed for that path. ACK messages are only accepted if they arrive after the server has issued a RSP, otherwise they are ignored.

5.5.2.5.8 CANCEL (CCL)

A CANCEL message can be sent by PCS clients and PCS servers. CCL messages can be generated during the normal path computation cycle but also in case of an abnormal termination of a PCS to PCS communication.

If a PCS, acting as a server for the PCP session, received a CCL message from the PCS client, it must form new CCL messages and forward a CCL message to each PCS to which it sent a REQ for which it did not received any positive or negative reply. Once this has been achieved it must delete all its internal states referencing the PATH-COMPUTATION-ID indicated in the message. If the PCS has no pending request concerning this PATH-COMPUTATION-ID, it can optionally query its management plane to retrieve a possible existing contract referenced by this PATH-COMPUTATION-ID and delete it. Just before deleting this contract, it can form a new CCL message and forward it to the next PCS in the path. If it does not, the VALIDITY-DATE will be applied.

The same procedure applies if the PCS server detects a communication problem with one of its Client PCS. In that case, the PCS server issues CCL messages for all pending request received from this client PCS.

When a PCS, acting as a client for the PCP session, received a CCL message from a PCS server, this indicates that a PCS along the path towards the target destination has experienced communication problems leading to close a PCP communication. In such a case, each client PCS cancels all the internal states referencing this PATH-COMPUTATION-ID and forward this indication to the upstream client PCS up to the initial requestor.

5.5.2.5.9 State diagram

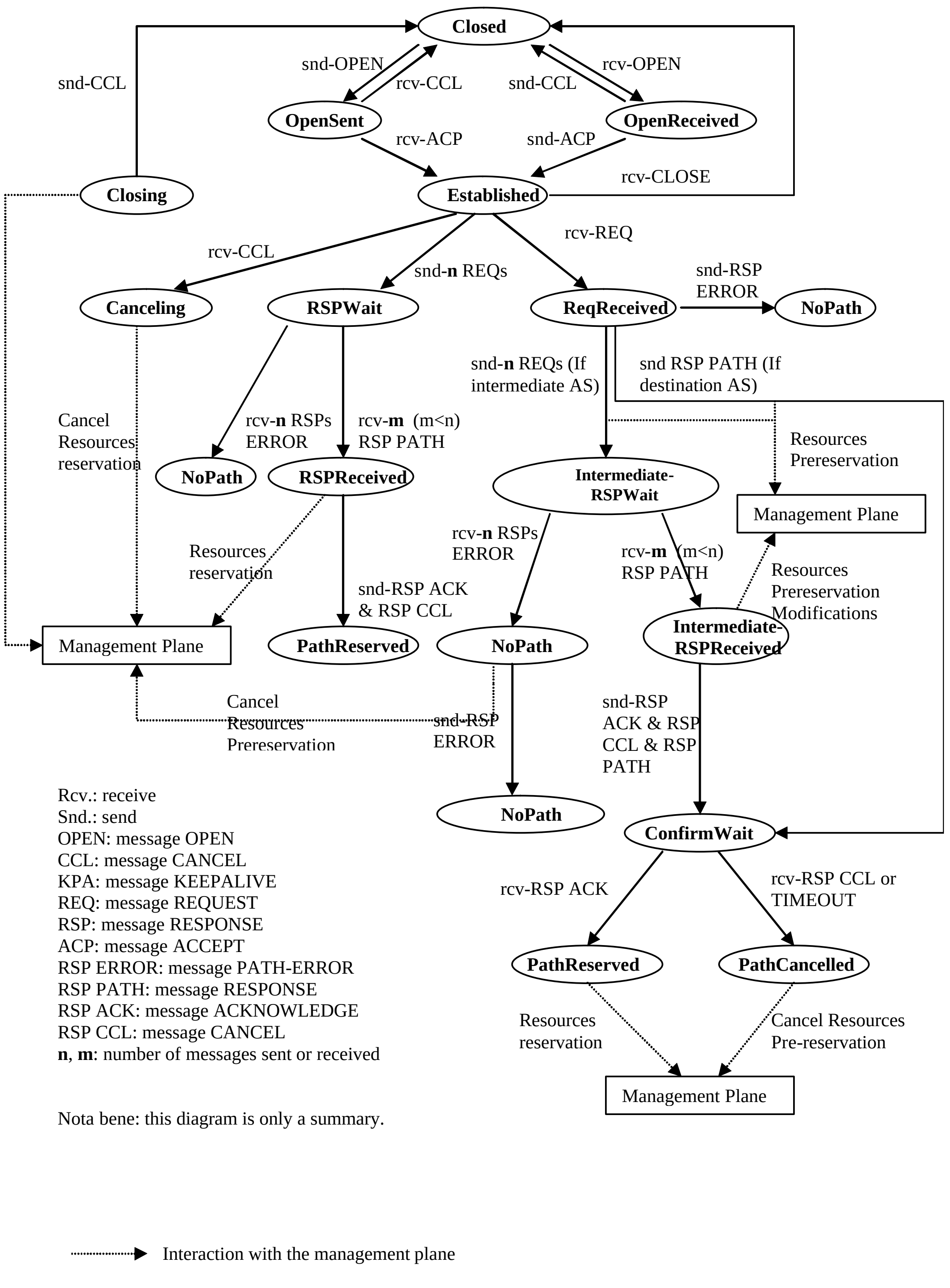
This state diagram (Figure 65) illustrates main state changes, together with the causing events and resulting actions. It is just a summary of the main events and does not include all the possibilities. For example, in the diagram there is no event that leads to the closing state but there are a lot in reality. Meaning of these states is hereafter briefly described:

- *Closed*: a TCP connection exists but no PCP session.
- *OpenSent*: PCP is waiting for an ACP message.
- *OpenReceived*: PCP need to send an ACP message.
- *Established*: represents an open connection. REQ and RSP messages can be exchanged.
- *Closing*: the peer asks for closing the session but the current state is not *established*. That means that the request has not finished. The PCS must cancel the pre-reservation (if any) and advertise the others PCS concerned by the path computation.
- *RSPWait*: represents the source PCS waiting for all the response of the request.
- *Intermediate-RSPWait* : represents an intermediate PCS waiting for all the response of the request.
- *ReqReceived* : When receiving a REQ message, the PCS tries to find a local path that fits the constraints of the request. If none exists, no path can be found for the REQ. If one or more are found, the PCS must pre-reserve the resources for those paths. If it is an intermediate PCS, it must send request to the possible next PCSs.
- *NoPath*: it is a fictional state, it is just used to indicate the fact that no path could be found and that there is no more actions to do (corresponding to the *established* state).
- *RSPReceived*: represents the reception of one or more RSP message. In this case, the PCS must choose one or many path. It must ACK the chosen paths and cancelled the others (that were prereserved).
- *Intermediate-RSPReceived* : the treatment of this state is the same as the *RSPReceived state* but it must advertises the chosen paths to the upstream PCS too.
- *PathReserved*: it is a fictional state, it is just used to indicate that a path has been found and reserved. The PCS has nothing else to do. The use of this LSP will be negotiated by the management plane.
- *ConfirmWait*: represents waiting for the ACK of the path proposed to the upstream PCS. When the ACK is received, the resources are administratively reserved for a certain amount of time which should be long enough to allow the final negotiation via the management plane.
- *PathCancelled*: When a path is cancelled or has not been ACKed, the pre-reservation is cancelled.

5.5.2.6 Test objectives

The objectives of the tests are to verify that the Path Computation Server specification provides MESCAL with the requested functionality. In particular testing specification and activities shall verify that:

- The PCS can find inter-domain paths satisfying various QoS constraints.
- The PCS can optimise the QoS guarantees attached to path.
- Concept and algorithms are reliable and scalable:
  - Convergence time
  - Loop avoidance
  - Prevent deep innondation



Rcv.: receive
Snd.: send
OPEN: message OPEN
CCL: message CANCEL
KPA: message KEEPALIVE
REQ: message REQUEST
RSP: message RESPONSE
ACP: message ACCEPT
RSP ERROR: message PATH-ERROR
RSP PATH: message RESPONSE
RSP ACK: message ACKNOWLEDGE
RSP CCL: message CANCEL
n, **m**: number of messages sent or received

Nota bene: this diagram is only a summary.

Figure 65. PSP state diagram

5.6 IP-based Intra-domain TE

5.6.1 Introduction

5.6.1.1 Overall Objectives

The purpose of intra-domain Traffic Engineering is to configure the intra-domain network in such a way that it satisfies the requirements of the traffic forecast. The forecast provides Intra-domain Traffic Engineering with demands for ingress, egress pairs and QoS constraints. The Intra-domain Traffic Engineering functional block is responsible for the distribution of this traffic among the available network resources as efficiently as possible while honouring the given QoS constraints.

The functional architecture diagram in Figure 66 highlights the *Offline Intra-domain Traffic Engineering* block.

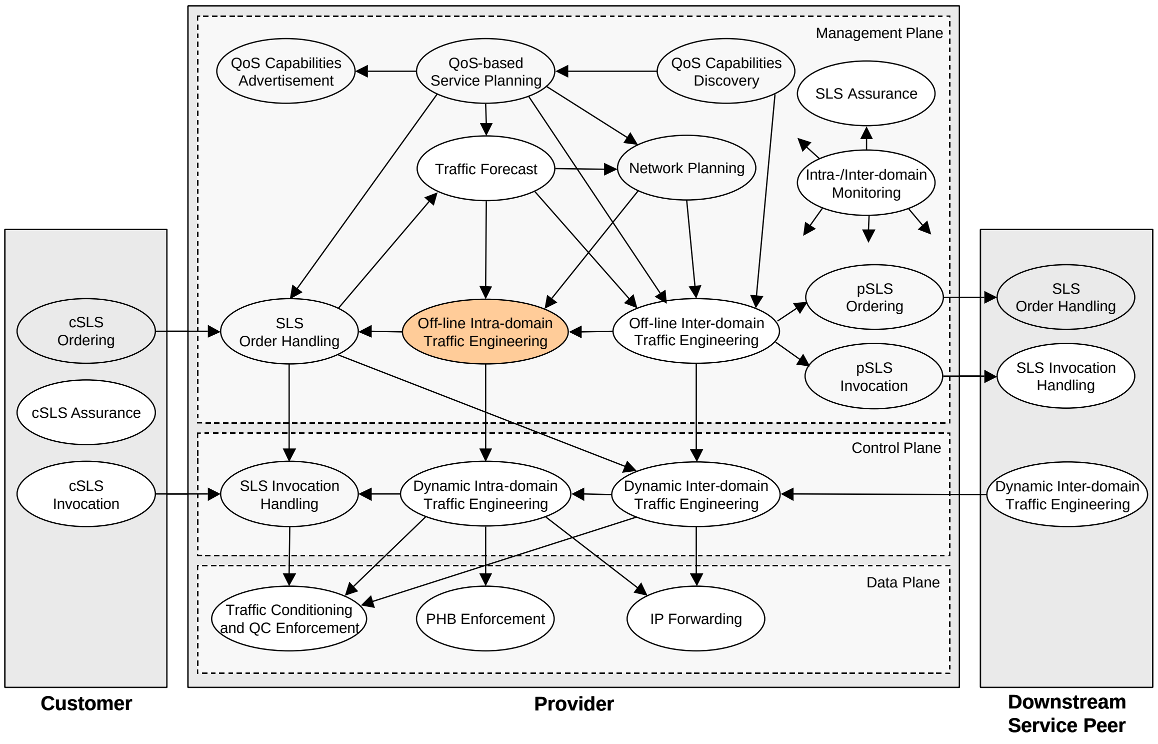

Management Plane
QoS Capabilities Advertisement
QoS-based Service Planning
QoS Capabilities Discovery
SLS Assurance
Traffic Forecast
Network Planning
Intra-/Inter-domain Monitoring
cSLS Ordering
SLS Order Handling
Off-line Intra-domain Traffic Engineering
Off-line Inter-domain Traffic Engineering
pSLS Ordering
SLS Order Handling
cSLS Assurance
pSLS Invocation
SLS Invocation Handling
Control Plane
cSLS Invocation
SLS Invocation Handling
Dynamic Intra-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Dynamic Inter-domain Traffic Engineering
Data Plane
Traffic Conditioning and QC Enforcement
PHB Enforcement
IP Forwarding
Customer
Provider
Downstream Service Peer

Figure 66. MESCAL Functional Architecture, highlighting Offline Intra-domain TE

Ideally, the distribution of traffic on the available network resources is carried out without any implementation constraints. However, the IP based Traffic Engineering described in this chapter is based on optimising OSPF link weights. Link weight based shortest path routing is not as flexible as an optimal routing solution. This is because the routing is limited to the shortest path algorithm that is run individually by each router of the network. OSPF is using only the destination address and link weights to route packets, whereas more differentiated routing decisions would require more information. A mechanism allowing for more flexibility is MPLS, however, it comes at the cost of introducing state information into the network, as well as loss of the self management capability (routing) of IP networks.

The IP traffic engineering algorithm proposed in this chapter is based on the assumption that the flexibility of MPLS like solutions is not required to efficiently support quality of service, but that the efficient optimisation of OSPF link weights allows sufficient control.

In addition to the OSPF link weight traffic engineering, it is proposed to support one weight setting per DiffServ code point (DSCP) on each link. This is not to be confused with the DiffServ Per Hop Behaviour (PHB), which is a queuing behaviour defined by as part of the Differentiated Services [RFC2475]. The DSCP values can be mapped onto a PHB as well as serve other purposes, such as provide additional routing information. When considering the DSCP value as additional routing information, it is possible to route all traffic aggregates one DSCP "routing plane" differently to that of other DSCP routing planes.

The Offline Intra-domain IP Traffic Engineering block is also responsible for adapting the network to predicted changes in demand. These changes include an increase in demand during office hours and evenings. When considering different QoS enhanced services, changes in service may also occur during these hours, VoIP during office- and streaming Video during evening hours. Further possibilities are surges in demand, in the event of a public holiday where resources could be committed away from office applications. A different type of change could be the failure of an important network link, for which a backup configuration is available. In summary, offline Intra-domain TE can put into effect predicted, pre-computed scenarios. These pre-computations are carried out by the offline-TE during its idle times between Resource Provisioning Cycles. It is important to point out that offline intra-domain IP TE cannot manage sudden changes in demand or topology that where not previously predicted without the processing needed to compute a set of link weights. Therefore, while the link weight computation is not dynamic, the algorithm responsible for recognising the need for network adaptation and for effecting weight modifications is. As discussed in detail in section 5.6.2.3, real link weight computation is computationally expensive and so it is difficult to conceive of more dynamicity in the actual weight computation.

This study has two purposes. To improve the IP based intra-domain traffic engineering results from the IST-Tequila project and to take a fresh approach at the intra-domain routing problem in the light of the inter-domain traffic engineering developed in IST-MESCAL. Special emphasis is placed on the optimisation of interactions between inter-domain and intra-domain traffic engineering.

5.6.1.2 Decomposition of Functionality

Figure 67 shows the internal structure of the Offline Intra-domain Traffic Engineering block and its relationship with neighbouring blocks. There are two sub-components, *Resource Optimisation* and *Network Reconfiguration Scheduler*.

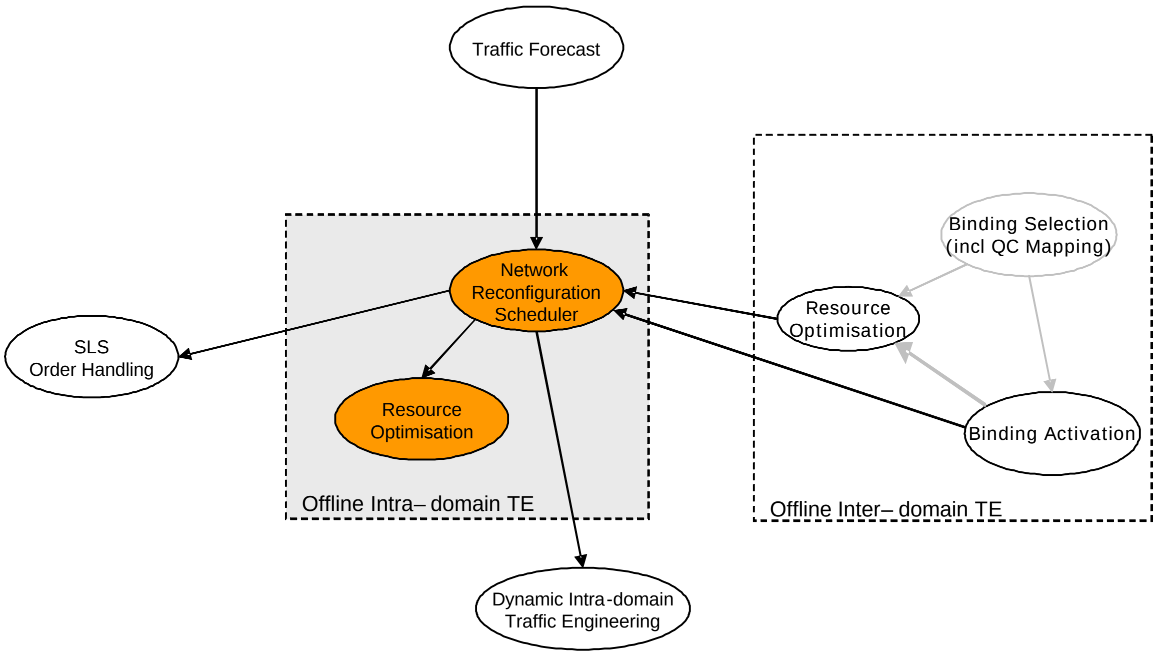


Figure 67. Decomposed Intra-domain Traffic Engineering

The *Resource Optimisation* block contains the OSPF link weight optimisation algorithm. It is a passive block, until called by the *Network Reconfiguration Scheduler* at which point it collects a traffic demand matrix and a network topology and computes an optimal set of link weights. Computed weights are deposited in a link weight database inside the Offline Intra-domain Traffic Engineering block, until they are put into operation in the network by the *Network Reconfiguration Scheduler.*

The *Network Reconfiguration Scheduler* is the control system for the Offline Intra TE block. It has two main purposes, handling computation requests to *Resource Optimisation* (Resource Provisioning Cycles, Inter-domain Traffic Engineering "what if" queries, etc) and scheduling the reconfiguration of the network using link weight settings computed by the *Resource Optimisation* block. Requests for network reconfiguration that have not been computed previously are passed to the *Resource Optimisation* together with information on where to retrieve traffic demand matrices and network topologies for the link weight computation. The *Network Reconfiguration Scheduler* uses its scheduling capabilities in order to predict (or be alerted to) the periodic changes in demand, for which pre-computed network configurations are available in the link weight database. When the scheduler is alerted to yet unknown changes in demand or topology it invokes *Resource Optimisation*, the results of which are implemented in the network but are also stored for possible reoccurrences in the future. This way the scheduler learns about the dynamic behaviour of the network, improving its effectiveness with time.

5.6.2 Resource Optimisation

5.6.2.1 Objectives

- Convert *Traffic Forecast* into OSPF link weights, while honouring QoS constraints and optimising network capacity.
- Provide "what if" scenarios to *Offline Inter-domain Traffic Engineering* in order to optimise inter- intra-domain interactions.
- Provide network configuration scenarios to *Network Reconfiguration Scheduler* for pre-computation.

5.6.2.2 Interface Specification

Figure 68 enlarges the decomposed Offline Intra-domain Traffic Engineering, providing a clear definition of interfaces required internally.

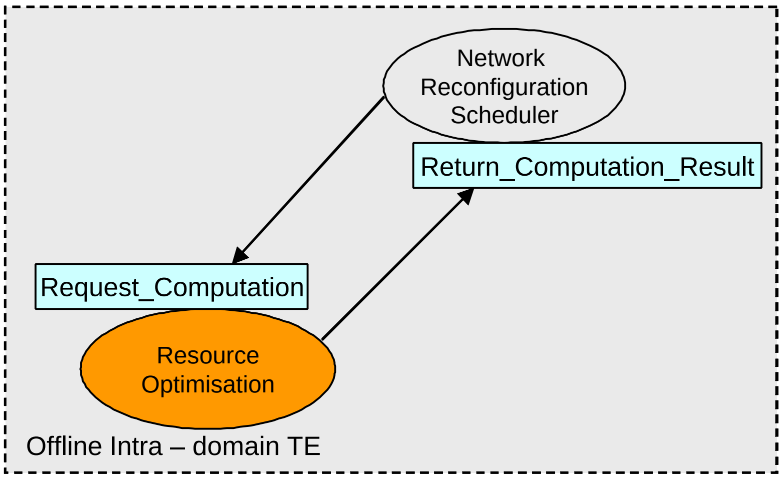

Network Reconfiguration Scheduler
Return_Computation_Result
Request_Computation
Resource Optimisation
Offline Intra – domain TE

Figure 68. Interactions of the Resource Optimisation block

- **Resource Optimisation & Network Reconfiguration Scheduler**

  Request_Computation(handle to iTM)

  The Network Reconfiguration Scheduler requests the computation of link weight optimisation for a particular network topology and demand matrix. Location of both topology and demand matrix have to be specified by the Network Reconfiguration Scheduler at the time of the request (locations for these may be databases of *Traffic Forecast* or databases for "what-if" scenarios inside *Offline Intra-domain Traffic Engineering*).

  Return_Computation_Result()

  Once the Resource Optimisation has completed the request, it stores the information in the link weight database and notifies the Network Reconfiguration Scheduler.

5.6.2.3 Algorithm Description

5.6.2.3.1 Motivation

Distributing traffic demands on OSPF networks is a challenge, because of the indirectness of the traffic engineering problem. Finding optimal paths for each traffic demand as for MPLS is only half of the problem, the other half is the implementation of these paths into OSPF link weights. While MPLS allows the explicit pinning of a route between any two nodes and effectively switches packets according to this configuration, OSPF relies on individual routing decisions taken at each node. These are based solely on destination IP address and the networks link weight metrics that determine the shortest path towards the destination address. It is therefore clear that mapping an "optimal" demand distribution as calculated for an MPLS network onto an OSPF network is not an achievable goal in most practical cases. Instead, the algorithms for finding good paths and for translating these paths into link weights, have to go hand in hand, iteratively searching for better link weights to spread the load across the network.

Several link weight optimisation techniques to achieve this have been proposed. There are heuristic, genetic and hybrid (memetic) algorithms for solving the link weight computation. Most proposed algorithms follow slightly different optimisation criteria, minimum average utilisation, maximised capacity or a combination of those and other criteria weighted in a cost function. A much cited link optimisation algorithm was developed by Bernard Fortz and Mikkel Thorup and has been published in [Fortz00] with several iterations and improvements in [Fortz02a, Fortz02b and others]. The algorithm aims to maximise the networks free capacity and minimise the number of heavily congested links. Several experimental studies followed, aimed at improving the heuristic weight setting algorithm described in [Fortz00]. Algorithms described in [Eric02, Buriol, Riedl] are all exploring genetic and memetic heuristics.

All IP link weight optimisation algorithms discussed so far are aimed at some form of network utilisation optimisation. None are explicitly designed to satisfy quality of service constraints of the traffic. The algorithms designed as part of the IST-Tequila [TEQUI,D1.4] project are specifically targeted at traffic engineering in QoS aware networks and support DSCP aware routing.

5.6.2.3.2 Optimisation Algorithm Objectives

- The primary objective is to distribute the demand projected by *Traffic Forecast* in such a way that all QoS constraints are honoured for as long as the demands do not exceed their specified bandwidth.
- In addition to the primary objective it is important to maximise the available capacity within the network, while ensuring that this capacity is well balanced across the network so that arising demands can be satisfied without extensive reconfiguration of the link weights. Thus there is a trade off between balancing and optimising free capacity.

5.6.2.3.3 Link Weight Computation Algorithm Outline

This section defines the basic organisation of the algorithm and then discusses each step in more detail depicting the operation.

1. **Initialise the network** by setting some arbitrary (e.g. unit value) weights for each link and each DSCP routing plane. This defines the initial condition of the weight setting algorithm.
2. **Calculate the shortest path tree** for each ingress-egress pair that has a traffic load. This has to be repeated for each DSCP routing plane. Check the hop count limits that each routing plane has to obey.
3. **Calculate the total link load** for all links and hence identify the most congested link. Each DSCP routing plane has to be taken into account individually, although the total link load is a single integer figure representing total effective link load across all planes. Traffic for each PHB constitutes differently to the effective link load.
4. **Reduce the load on the most congested link.** This is the most crucial task of the algorithm and its many options and tradeoffs will be discussed in detail.
5. **Re-compute steps 1 to 5** and test to see if step 4 was successful. If unsuccessful, ban the solution and compute steps 1 to 4 again, if successful recomputed steps 1 to 4 until no more improvement can be achieved or until the computation is stopped.

5.6.2.3.4 Routing Model Definitions

In order to define a weight manipulation algorithm and test its performance, it is essential to set up some definitions. As in [RFC2702] the network is modelled as a directed graph, $G = (N, E)$ where the nodes $n \in N$ and links $l \in E$ represent routers and links between routers. A link *l* has capacity *c(l)* indicating the amount of traffic that *l* can accommodate. The traffic is given in form of a demand matrix *D*, representing the amount of traffic flowing on a path between any nodes (s, d). This demand matrix is provided by traffic forecast, which in turn is calculated based on historic data and intra/inter-domain traffic demands. It can be expected that many of the demands (s, d) are zero, as not every ingress/egress pair has traffic flowing between it. So the routing problem is to distribute the traffic from non zero $D(s, d)$ across the network evenly. The load on a link is given as x_l, this is the sum of all demands $D(s, d)$ using the link *l*. The utilisation of *l* is then given by $x_l / c(l)$. For the weight allocation, a weight *?* is assigned to each DiffServ Code Point $h \in H_l$ on each *l*, so that $w_{h,l}$ denotes a unique weight.

5.6.2.3.5 QoS Constraints

The algorithm has to meet the QoS constraints of each flow. Constraints of delay, jitter and packet loss probability are imposed on each Per Hop Behaviour and are therefore the same throughout a single routing plane.

- In order to ensure an upper limit on queuing delay, a maximum queue length has to be defined per PHB. The queue length limit is enforced by dropping excess packets and so both queuing delay and the arising jitter can be enforced by imposing a **hop count constraint.**
- Similarly, it can be demonstrated (e.g. in [TEQUI,D1.4]) that packet loss probability and the end-to-end delay can be seen as a hop count constraint problem, if some simplifying assumptions are made and link loads do not exceed the planned amounts.

- As the queue length has an upper bound, packet loss has to be accommodated through distribution of link loads and prevention of congested links with the link weight calculation algorithm. This can only be ensured if the traffic demands do not exceed the ones specified in the *Traffic Forecast* that was used for the link weight calculation. Some degree of excess link load may be tackled by over specifying the demands for the link weight calculation. However, an admission control policy is unavoidable for quality of service enforcement, to police that the traffic does not exceed the planned capabilities of the network.
- Since hop count constraints do not currently feature in the routers shortest path algorithms, the hop count limit has to be enforced by the IP Traffic Engineering algorithm. The constraint on hop count also introduces a limitation of the OSPF based traffic engineering. All traffic flowing from one ingress point to the same egress point that flows on the same routing plane, must follow the same route through the network. It follows that the hop count for all this traffic is the same (with the exception of equal cost multipath (ECMP) [RFC2328], which will be discussed later). A further complication to this problem becomes apparent when taking into account that all traffic from any ingress to the same egress, that is on the same routing plane, might merge into an aggregate flow at some point. If this is the case, then hop count constraints from all these ingress points to the same egress has to be taken into account when modifying the route of such an aggregate flow.

5.6.2.3.6 Initialising the Network

This step is important, because it defines the initial conditions for the weight setting algorithm. Better initial conditions should lead to faster and better convergence. The apparent choices are random, unit or inverse capacity weight settings. In operational networks, inverse capacity weights are often used. Their advantage is that OSPF now automatically favours the larger links towards the core of the network and so inverse capacity link weights are a crude but practical approach to traffic engineering. It has been shown in [Fortz02a] that initial weight settings based on inverse link capacity, greatly improve the convergence time of the algorithm.

5.6.2.3.7 Calculating the Shortest Path Trees

Principally this needs to be done for each load bearing ingress egress tuple. However, because each DSCP routing plane has a unique set of link weights, the shortest path tree has to be calculated for each ingress egress tuple and for each DSCP thus multiplying the routes that have to be calculated by a maximum of 64. It is essential that the algorithm for calculating the path is exactly the same as that of the routers in the network, since all weight setting is based on these paths. After performing this step, hop count constraints have to be confirmed for each *D(s,d)*. If a previous iteration of the algorithm has caused a hop count limit to be exceeded, the solution has to be discarded.

5.6.2.3.8 Calculating the Total Link Cost

The total link load is calculated by distributing the traffic from each ingress egress tuple over the shortest paths calculated in the previous step. Link loads from each routing plane and the physical capacity of each link are used to give the total free capacity of each link. Each PHB has a different delay and loss probability which has to be expressed by associating an *equivalent* bandwidth to PHBs rather than the physical bandwidth. For each PHB h of a set H_l of PHBs on a link with bandwidth allocation $x_{l,h}$, the equivalent bandwidth can be expressed as a function $f_{l,h}(x_{l,h})$ increasing in $x_{l,h}$ and greater for any given $x_{l,h}$ with higher priority h. The total equivalent load of each link is then $\sum_{h\in H_l} f_{l,h}(x_{l,h})$. Assuming that $c(l)$ is not the same for all links,

$$L_e = \sum_{h\in H_l} f_{l,h}(x_{l,h}) \Big/ c(l)$$

is the normalised utilisation of the link.

In order to arrive at an overall cost function, the statement has to be extended to reflect a cost per link which can then be summed over $l \in E$.

$$\Phi = \sum_{l \in E} \Phi_l(L_e) = \sum_{l \in E} \Phi_l \left(\frac{\sum_{h \in H_l} (f_{l,h}(x_{l,h}))}{c(l)} \right)$$

The cost function should be convex and increasing to avoid highly congested links. The actual function Φ_l could be approximating an exponential curve with discrete values as defined in [Fortz00] but the curve should rise more rapidly in order to avoid congested links to preserve the QoS packet loss probability.

5.6.2.3.9 Reduce the load on the most congested link

In order to minimise the cost function, those links with the largest contribution to the total cost have to be identified. Hence $l_c = \max_{l \in E} \Phi_l(L_e)$ identifies such a link. Before modifying any weights, it is necessary to choose a candidate traffic flow passing though l_c which to modify. Depending on the type of modification and the amount of load, it may be good to choose either a small or a large flow. There are several DSCP routing planes to select from. As long as the initial condition of the algorithm is based on a good weight setting (i.e. inverse capacity or a good previous result), it may be a good option to leave traffic mapped to high priority PHBs and modify first the lower priority traffic. This ensures that high priority traffic stays on the large capacity links and it also guarantees least disruption of this traffic in case of a traffic engineering event such as a Resource Provisioning Cycle (see Section 5.6.3.3.1). The high priority traffic will not be as affected if its paths do not change.

The choices made here on DSPC routing plane and size of traffic flow represent a small subset of the available options, detailed simulations of the IP Traffic Engineering system should reveal the most suitable choices.

Let R be the origin node of link l_c, and let R_l be the destination node of l_c. Let *W(R)* be the sum of the weights on the shortest path from *R* to some egress node R_E. Also let R_n be a neighbouring node to *R* of a set of neighbours $R_n \in B$. Then,

$$W(R) = W(R_l) + w_{l,h} \leq W(R_n) + w_{(R,R_n),h} \text{ for all } R \in B$$

The objective now is the reduction of the load on l_c, by redistribution of its load onto other neighbouring links.

Modifying a single link weight.

For this single weight modification, the neighbourhood of node R is searched in order to locate a neighbouring node R_n such that

$$W(R_n) + w_{(R,R_n),h} + j < W(R_l) + w_h \text{, where } j \text{ is a weight adjustment}$$

It is sensible to choose the neighbour where j is the smallest value in all of *B*. The reason is that the adjustment of a weight may cause other routes on R_n to change. By choosing the smallest weight change, the probability of such unwanted changes is kept low.

Balancing the loads.

Large aggregated flows develop naturally in networks with shortest path algorithms, as a result of traffic aggregation from multiple ingress points towards a particular egress point. As soon as these different traffic flows meet on a node, they become a single aggregate flow. A network consisting of such large aggregates is difficult to optimise. However, it is possible to split these flows between neighbouring nodes by making use of equal cost multi path which causes a split of the traffic when more than one shortest path route is available. By setting the *W(R)* equal to some or all of $W(R_n) \in B$ so that all those R_n lie on the shortest path. In order to do this,

$$W(R_1) + w_{(R,R_1)} = W(R_2) + w_{(R,R_2)} = \dots = W(R_n) + w_{(R,R_n)}$$

should all be set equal. It is important that

$$W(R_n) + w_{(R,R_n)} \le W(R_y) + w_{(R,R_y)}, \text{ where } R_n \in B, R_y \notin B$$

holds true for this modification, as a reduction in weights may cause loading of a link that is already heavily loaded. The drawback of this approach is that a carefully set up multi path can be damaged when further iterations of the algorithm modify paths nearer the sink of the ECMP traffic. These modifications could cause for some of the paths to be no longer on the shortest path, or for one of them to become shorter than all others and attracting all of the traffic. Care has to be taken in order to prevent these scenarios from occurring by e.g. applying the ECMP rules only in the last iterations, or as a last resort. It could also be conceived of keeping a database with all the ECMP weight changes to remember which links have been modified.

Some undesirable consequences of link weight modifications.

Although modifications made to link weights should, in general, succeed at redistributing the traffic flow in question, they may have undesirable side effects. A full view of changes that a link weight modification has caused, becomes visible after the shortest path algorithm has been run on the new set of weights and costs have been calculated for each link. Two problems may occur:

- Traffic shifted away from one link has caused congestion on a neighbouring link.
- The route modification has caused the hop count constraint of the rerouted traffic to be exceeded. This could be the case for the traffic that was chosen for rerouting and also for other traffic that was rerouted though the weight change.

If neither of the two cases holds, the weight modification can be accepted and the algorithm continues. If a problem occurs, the weight modifications have to be discarded (and also banned from being chosen again) and other, different modifications have to be identified to redistribute the traffic.

5.6.2.3.10 Re-computation of steps 1 to 5

The iteration of the algorithm has two purposes, to check on the effectively of the changes made and to continue making changes if necessary. Re-computation should continue until no further improvement can be made or until a certain number of iterations has passed. The number of iterations could also be controlled by a cost function analysing the time taken to achieve further optimisation. A threshold could then be defined to stop the algorithm when improvement is too slow.

5.6.2.3.11 Anticipated Problems

It is well known that the weight optimisation algorithm is computationally expensive and this dilemma is reflected in several mechanisms to reduce the complexity of computation. To the authors knowledge, this is the first algorithm based on [Fortz00] to assume weight settings for multiple routing planes and it is therefore unclear how much extra load this will cause to the algorithm. Assuming some techniques can be applied for localising the link weight search during the implementation phase as well as considering the available processing power in recent years, an optimistic view can be taken towards the computation time.

5.6.2.4 Test requirements

The Resource Optimisation algorithm will be tested to show that it correctly distributes the demand matrix on the network, while satisfying the QoS constraints.

- Algorithm stability
    - convergence time
    - no looping (especially with ECMP)
- Algorithm scalability
    - how network size affects convergence time
    - are DSCP routing planes required or is one routing plane sufficient
- Correct free capacity distribution with no overloaded links,
    - calibration of link weight cost function
- Reliability of algorithms compliance with hop count constraints
- Run time calibrations, testing the number of iterations needed to arrive at result
    - calibration of the termination cost function
- Studies on algorithm diversification techniques
    - increasing the search space
    - arrival at better minimum solutions
- Studies on techniques for few weight changes to arrive at new optimum solutions

5.6.3 Network Reconfiguration Scheduler

5.6.3.1 Objectives

- Dynamic reconfiguration of network configuration to adapt to changes in
  - traffic demand
  - network topology
- Reconfiguration according to schedules
- Effecting the network configuration of the offline Intra-domain Traffic Engineering

5.6.3.2 Interface Specification

Figure 69 enlarges the relevant parts of the architecture, in order to allow a clear definition of interfaces required.

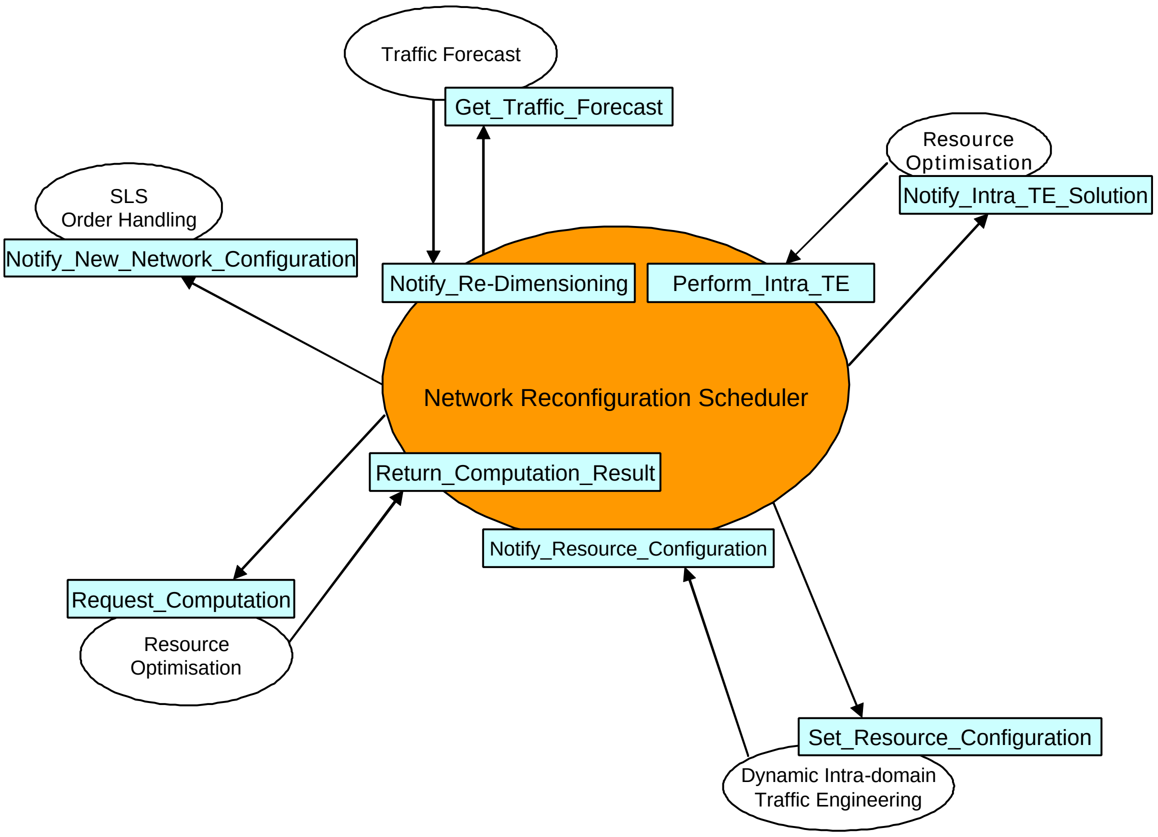

Traffic Forecast
Get_Traffic_Forecast
Resource Optimisation
Notify_Intra_TE_Solution
SLS Order Handling
Notify_New_Network_Configuration
Notify_Re-Dimensioning
Perform_Intra_TE
Network Reconfiguration Scheduler
Return_Computation_Result
Notify_Resource_Configuration
Request_Computation
Resource Optimisation
Set_Resource_Configuration
Dynamic Intra-domain Traffic Engineering

Figure 69. Interactions of the Network Reconfiguration Scheduler Block

- **Traffic Forecast & Network Reconfiguration Scheduler**

  Get_Traffic_Forecast(handle to iTM)

  This method is called by *Resource Reconfiguration Scheduler* in order to request information on where to locate (inside a database) the current traffic forecast. (Note that *Resource Optimisation* will be passed the information so that it can retrieve the forecast for computation.) This method can also be used to retrieve a computed hypothetical iTM' for an inter-domain TE "what if" scenario

Notify_Re-Dimensioning()

Traffic Forecast will call this method in order to notify *Resource Reconfiguration Scheduler* that the current network configuration is not satisfactory to accommodate the demands and that a Resource Provisioning Cycle is needed.

- **Network Reconfiguration Scheduler to SLS Order Handling**

  Notify_New_Network_Configuration(iRAM)

  This method will be called by *Resource Reconfiguration Scheduler* in order to notify *SLS Order Handling* that a new network configuration is available.

- **Network Configuration Scheduler & Dynamic Intra-domain Traffic Engineering**

  Notify_Resource_Configuration()

  This method will notify the *Network Reconfiguration Scheduler* that the dynamic resource management was not able to accommodate the demand with its current configuration or that link failures have occurred.

  Set_Resource_Configuration (Link weight matrix w_o *)*

  This method will be called by *Network Reconfiguration Scheduler* passing the scheduling information, link weights and ECMP settings to be configured for each router order to affect a change in the network.

- **Inter-domain Resource Optimisation & Network Configuration Scheduler**

  Perform_Intra_TE (handle to iTM')

  This method will be called by inter-domain *Resource Optimisation* to consult *Network Configuration Scheduler* on the resulting intra-domain resource availability and utilisation if a given inter-domain TE solution is activated. For this, a handle to a modified iTM is passed that includes the additional, hypothetical inter-domain demands.

  Notify_Intra_TE_Solution (iRAM', cost F*, (full intra-domain TE configuration))*

  This method will be called by *Network Reconfiguration Scheduler* to return the intra-domain traffic engineering solution or resource availability (intra-domain configuration, e.g. iRAM') to inter-domain *Resource Optimisation*. In addition, a cost for network reconfiguration is passed should the presented solution be implemented. At this stage, it is left for further study to determine whether the iRAM provides sufficient information to the inter-domain TE or whether the full intra-domain configuration needs to be passed.

- **Binding Activation to Network Configuration Scheduler**

  Notify_Inter_TE_Solution (handle to selected solution)

  This method will be called by *Binding Activation* to indicate to off-line intra-domain TE which inter-domain TE solution (i.e. eRAM$\in$eRAM(s)) has been selected. The purpose of this notification is to enable the *Network Configuration Scheduler* to physically configure the network resources, thereby enabling the selected resource allocation.

- **Resource Optimisation & Network Reconfiguration Scheduler**

  Request_Computation(handle to iTM)

  The Network Reconfiguration Scheduler requests the computation of link weight optimisation for a particular network topology and demand matrix. Location of both topology and demand matrix have to be specified by the Network Reconfiguration Scheduler at the time of the request (locations for these may be databases of *Traffic Forecast* or databases for "what-if" scenarios inside *Offline Intra-domain Traffic Engineering*).

Return_Computation_Result()

Once the Resource Optimisation has completed the request, it stores the information in the link weight database and notifies the Network Reconfiguration Scheduler.

5.6.3.3 Algorithm Description

5.6.3.3.1 Modifying link weights in an operational Network

There are problems associated with the modification of link weights in an operational networks. Each single link weight update has to be flooded as a link state advertisement which causes re-computation of the routers forwarding tables. This is a disruptive process with the amount of disorder introduced into the network being a function of the number of modified link weights. It is therefore desirable to modify as few link weights as possible, as few times as possible. An example for a technique to limit the number of weight changes at each iteration is presented in [Fortz02a] as an extension to the link weight setting algorithm. The disruption caused by link weight modification, is a deterrent from optimising to frequently and careful considerations have to precede an optimisation decision.

There are two different causes for network optimisation to become necessary: Intra-domain Resource Provisioning Cycles and dynamic events that occur on smaller timescales, within an Intra-domain RPC.

Resource provisioning cycles of the intra-domain and inter-domain TE always coincide and can cause extensive reconfiguration of the network according to long term changes in demand. However, it is also possible for a resource provisioning cycle to cause little reconfiguration in the intra-domain TE, if the demands can still be satisfied with little or no link weight modification. Although a more optimal solution may be found through an extensive reconfiguration, it is not always carried out because of the additional cost of network disruption. Network disruption can therefore be expressed as a threshold cost, causing the network to stay in a suboptimal state until reconfiguration is unavoidable. The value of the threshold is proportional to the size of the network (time to convergence after disruption) and number of link weight modifications necessary (amount of disruption introduced). Finding solutions with as few weight changes as possible is therefore beneficial.

Smaller, dynamic changes in demand that occur within the Intra-domain RPC (as outlined in the next section) have to cause minimal disruption to the network and are based on one or few link weight changes. This method will be applied to optimise for new concentrated demands like pSLS traffic as well as link failures, etc. Fluctuations in daily demand could also be addressed in this way, but it may be beneficial to take these into account by optimising weight settings for multiple demand matrices, because this technique does not require any periodic weight changes. The technique aims at selecting a weight setting solution that is good for a set of demand matrices that reoccur periodically. Further studies on the dynamic events occurring within the RPC, will be studied in more detail in the course of the project.

5.6.3.3.2 Dynamic events

The *Network Configuration Scheduler* is responsible for effecting link weight modifications that have already been computed. A small subset of events that can be foreseen in this way are

- Changes in demand
    - yearly demand changes
        - public holidays
        - new applications
    - weekly demand changes
        - weekends, public holidays
    - daily demand changes
        - office hours, etc

  - extraordinary events
    - Christmas, new years eve (this does not include unforeseen disaster events such as earthquakes, although some of its implications could be foreseen and effected in an emergency case)
- Changes in Network topology
  - Link failure
  - New Links

These events with their different time scales can be used to produce demand matrices for the *Network Configuration Scheduler* which has two functions.

- Schedule events to be pre-calculated by the offline Intra-domain Traffic Engineering
- Effect link weight changes when they become necessary (RPCs)

Some simple algorithms for the dynamic traffic engineering component are shown below, these are straight forward and do not require further explanation.

```
/*link failure
if link failure or scheduled event signalled do
    check database for pre-computed configuration
        if database entry exists do
            effect link weight modification
        else
            call offline intra-TE RPC
            add resulting configuration to database
        end
end
/*RPC event
if resource provisioning cycle completed in offline intra-domain TE do
    effect link weight modification
end
```

Given time, the *Network Configuration Scheduler* accumulates a large database of ideal network configurations for many scenarios. One could conceive of more sophisticated learning algorithms that accumulate useful network configuration scenarios more quickly. However, all scenarios become outdated when a large change in network topology occurs. While this might only have an effect when high capacity links are added or removed, it means that the *Network Configuration Scheduler* needs to adapt its scenarios if possible and discard them if they cannot be adapted. It may be possible to re-optimise an “old” network configuration by passing it to *Resource Optimisation* together with the new topology. This should consume less time than re-computing from scratch, although this is dependent on the extent of the changes. At this point the problem is left for further study.

5.6.3.3.3 Discussion of the interactions with the Inter-domain Traffic Engineering

The *Inter-domain Resource Optimisation* function block queries the *Resource Configuration Scheduler* for “what-if” scenarios in order to allow a choice of the best pSLS options not only based on inter-domain cost considerations, but also intra-domain resource availability and reconfiguration cost.

The *Inter-domain Resource Optimisation* passes information to *Traffic* Forecast that allows it to calculate and pass to intra-domain *Resource Optimisation* a projected iTM' for each "what if" scenario, including the additional inter-domain demands. The return value from the intra-domain *Resource Optimisation* should consist of a cost Φ and optionally an intra-domain configuration (e.g. iRAM'). Two scenarios are possible,

- The *Resource Configuration Scheduler* finds that the projected iTM' passes the threshold for causing a resource provisioning cycle. The intra-domain *Resource Optimisation* is executed, and the resulting cost and iRAM' is passed back to inter-domain *Resource Optimisation*. The cost function, should include the extra cost for necessary network configuration.
- If the intra-domain *Resource Configuration Scheduler* finds that the threshold is not crossed, it will return the iRAM' and the difference in cost between this projected iTM' and the iTM that the last resource provisioning cycle was based upon. Although the cost of all current network disorder is passed this way, it should be significantly less than a case involving a new weight setting from the intra-domain *Resource Optimisation*. This is because of the cost raised for disrupting the network configuration. This low cost raises the question if such "what if" scenarios based on no network reconfiguration should be passed back at all, and if so, if they should be limited to low bandwidth or short term pSLSs. It would be highly undesirable if a suboptimal pSLS is chosen on the grounds of a low intra-domain cost, an advantage that will disappear by the time the next resource provisioning cycle takes place.

5.6.3.4 Test requirements

The *Resource Reconfiguration Scheduler* has to be tested in order to show,

- Scheduling weight settings
    - Correct identification of the network state
    - Correct selection of pre-computed weight settings
    - Long term effectiveness of pre-computed cases for dynamic application, learning curve
- Correct simulations of states to be pre-calculated by *Resource Optimisation*
- Adaptations within the Resource provisioning cycle
    - Disruptiveness of few weight changes vs. less optimal network

6 TRAFFIC ENFORCEMENT

This Section presents the interactions and the behaviour of the data plane Functional Blocks (FBs) assumed by the MESCAL system: *Traffic Conditioning & QC Enforcement*, *PHB Enforcement*, and *IP Forwarding*. MPLS forwarding is also considered specifically for the hard guarantees solution option. The interactions of these FBs with the rest of the MESCAL FBs are defined. The behaviour specification of any FB explains the specification of the functions that are essential for the deployment of that specific FB.

6.1 Traffic Conditioning & QC Enforcement

6.1.1 Objectives

With the *Traffic Conditioning and QC Enforcement* function block we mean the process of realising the results of c/pSLS agreements and intra/inter domain TE functions in classifying, conditioning, and QoS class enforcement to traffic streams as appropriate to the o-QC treatment that these streams should receive per domain. These processes takes place at the data-plane after the c/pSLSs have been established, activated, and invoked. They are realised by downloading appropriate information for setting up the traffic classification and conditioning mechanisms to the DiffServ-capable routers. QC-signalling is performed across all domains using DSCPs/MPLS-EXPs. Traffic conditioning & QC enforcement must be performed:

At Network Edges (customer ingress point): Customer traffic at the edge routers should be classified in order to capture and reflect the negotiated *cSLSs*. In addition, suitable traffic profiles derived from the negotiated *cSLSs* should be enforced on the classified traffic before it actually enters the provider's network. The *SLS Invocation Handling* function block calculates the appropriate traffic classification and conditioning configuration parameters and downloads them to the Network Elements (NE) via the *Traffic Conditioning* function block. This block should also provide capabilities for statistics information retrieval with respect to the traffic classification and conditioning results.

The Dynamic inter-domain TE functional block provides configurations to the *Traffic Conditioning and QC Enforcement* function block for configuring the ingress NE to possibly perform DSCP re-marking for realising an intra-domain TE solution.

At the domain boundaries (ASBRs): Peer provider traffic at the border routers should be re-marked to the appropriate DSCP depending on the l-QC treatment it receives in the domain. The traffic should be classified in order to capture and reflect the negotiated *pSLSs* for traffic conditioning. In addition suitable traffic profiles derived from the negotiated *pSLSs* should be enforced on the classified traffic before it actually enters the domain. The *SLS Invocation Handling* function block calculates the appropriate traffic classification and conditioning configuration parameters and downloads them to the ASBRs via the *Traffic Conditioning* function block.

The *Dynamic Intra-domain TE* function block provides configurations to the *Traffic Conditioning and QC Enforcement* function block for configuring the egress ASBR to perform DSCP remarking for realising an inter-domain TE solution.

6.1.2 Interface Specification

6.1.2.1 Traffic Conditioning & QC Enforcement Interface to SLS Invocation Handling

The *Traffic Conditioning & QC Enforcement* (TC-QC) interface to *SLS Invocation* use basic concepts specified and documented in [RFC3290]. The model of different successive traffic conditioning elements contained in traffic conditioning blocks is adopted [RFC3289]. The output of each TC-QC element should be associated with the input of its subsequent element, which could be another traffic conditioning element. This way a full sequence of successive elements inside a TC-QC can be specified.

The *Traffic Conditioning & QC Enforcement* interface to *SLS Invocation* is defined as follows:

TC-QC_NewTC (TC-QC$_{ID}$, Interface, Direction)

Creates a new TC-QC to be applied in the ingress or the egress *Direction* of the specified *Interface*.

TC-QC_Classifier (ClassID, TC-QC$_{ID}$)

Creates a new classifier.

TC-QC_Filter (FilterID, ClassID, Precedence, Type, Parameters)

Creates a new filter as part of the classifier *ClassID*. The filter *Type* (BA, MF, other) with related *Parameters* Specific to *Type*. The *FilterID* will be applied to the *ClassID* with the *Precedence*.

TC-QC_Meter (MeterID, Type, Parameters)

Creates a new meter with meter *Type* that specifies an Average Rate, EWMA (Exponential Weighted Moving Average), Token Bucket, or other. The associated *Parameters* (e.g., average time interval) are specific to *Type.*

TC-QC_Marker (MarkerID, Type, Parameters)

Creates a new marker. The marker *Type* is DSCP/EXP and the related *Parameters* are specific to *Type*.

TC-QC_Shaper (ShaperID, Profile Parameters, Buffer Size)

Creates a new shaper. The *Profile Parameters* describe the profile the traffic is shaped to and the *Buffer Size* specifies the maximum queue length.

TC-QC_Dropper (DropperID, Type, Parameters)

Creates a new dropper. The dropper *Type* (WRED, other) and the related *Parameters* are *Type* specific.

TC-QC_GetStats (ElementID)

Returns statistics for *ElementID*. The type of statistics depends on the TC-QC element.

TC-QC_DeleteElement (ElementID)

Deletes the element identified by *ElementID.*

6.1.2.2 Traffic Conditioning & QC Enforcement Interface to Dynamic Inter- and Intra- domain TE

The *Dynamic Inter-domain TE* decision is enforced through configuration of the *Traffic Conditioning and QC Enforcement* function block, by configuring the ingress/egress ASBR to perform DSCP remarking. This is based on the fact that the TC-QC has been already created and TC-QC_Marker is called.

A similar interface as above is used for TC-QCs at the ingress point of a domain.

6.2 PHB Enforcement

This *PHB Enforcement* function block represents the required queuing and scheduling mechanisms for realising different PHBs associated with NE interfaces with appropriate configuration as determined by the related TE blocks. This block is responsible for implementing the mechanisms needed to provide differential forwarding treatments to traffic passing through the NEs based on the DiffServ specifications. The *PHB Enforcement* block manipulates the NE's native scheduling and queuing management mechanisms in order to enforce the parameters and the bandwidth and buffer sharing rules and policies, defined by the Dynamic Intra/Inter TE functional blocks, and hence satisfy the requirements in terms of throughput, delay, jitter and loss. *PHB Enforcement* should provide capabilities for PHB selection, PHB prioritisation, bandwidth and buffer resources allocation and excess resources sharing rules.

6.2.1 Interface Specification

Dynamic Inter-domain TE provides information for PHBs associated to the egress ASBR for realising inter-domain TE solution while *Dynamic Intra-domain TE* provides information for PHBs associated to the NEs within the domain for realising intra-domain TE solution. The interfaces of *PHB Enforcement* function block to these two function blocks are specified as below.

6.2.1.1 PHB Enforcement Interface to Dynamic Inter-domain Traffic Engineering

The following interface is used as *PHB Enforcement* interface to *Dynamic Inter-domain TE* for providing information to be used in ASBR type of NEs.

PHBEnf_NewPHB (PHBID, Interface ID, Name, Scheduling Class, Priority)

Creates a new PHB for an associated *Name* (e.g., EF) to be enforced in the interface identified by the *Interface ID*. The *Scheduling Class* parameter defines the scheduling class the PHB belongs to. The *Priority* parameter determines the priority with which the packets of this PHB will be served given that the resources allocated to each PHB are properly granted.

PHBEnf_MapDSCP (PHBID, DSCP)

Maps the *DSCP(s)* to the PHB identified by the *PHBID*. The packets marked with *DSCP* will be serviced by the *PHBID*.

PHBEnf_AllocateResources (PHBID, [Reserved Bandwidth], [Reserved Buffer], [Excess Bandwidth], [Excess Buffer], [Average Time, List of {Threshold, Dropping Probability}])

Allocates the scheduling resources to the *PHBID*. *Reserved Bandwidth/buffer:* the minimum amount of bandwidth/buffer is allocated to *PHBID*. *Excess Bandwidth/buffer:* the excess bandwidth/buffer can be used by PHBID in case of other PHBs' temporal under-use. *Average Time:* the time period over which the average queue size is calculated. A list of *Threshold* and *Dropping Probability* pairs determines the algorithmic dropping behaviour applied to *PHBID* for each virtual queue (i.e., WFQ, CBWFQ).

PHBEnf_ActivatePHB (PHBID)

Activate the *PHBID*. This activation results in downloading the PHB configuration parameters to the NE and enforcing the PHB.

PHBEnf_DeactivatePHB (PHBID)

Deactivates the *PHBID*. This deactivation results in releasing the PHBs allocated resources. These resources are still considered unavailable when the resources are checked for allocating to other PHBs.

PHBEnf_DelPHB (PHBID)

Deletes the *PHBID*. The allocated resources of *PHBID* are freed.

PHBEnf_GetStatistics (PHBID)

Returns the packets and bytes serviced by the *PHBID*, as well as the dropped packets and bytes for each defined *Threshold* and *Dropping Probability* pair.

A Similar interface is used as *PHB Enforcement* interface to *Dynamic Intra-domain TE* for providing information to be used in NEs within a network domain.

6.2.2 Behavioural Specification

6.2.2.1 Description of Functions

PHB Enforcement performs the four functions namely as Configuration, Verification, Enforcement, and Statistics. The Configuration function maintains the configuration and state information for each defined PHB. It is triggered based on the specific request for a configuration change. According to this request and the state of the PHB, it triggers the Verification and the Enforcement functions. The *PHB Enforcement* function block should check the requested resources against the available resources in the NE before granting resources to a PHB. The Verification function of PHB FB is triggered by the Configuration function. It calculates the available resources. It decides and its output is directed to the Configuration function as grant or rejection of requested resources for the PHB. *PHB Enforcement* should download the related information to the scheduling and buffer management mechanisms of the NE. An Enforcement function performs this task. When a configuration is successfully downloaded to the NE, the Enforcement function replies to the Configuration function, otherwise an enforcement error will occur. The Statistics function is for calculating statistics per PHB. This function inquires the NE, gathers the necessary information and then calculates the packets & bytes serviced and the packets & bytes dropped by the specific PHB.

6.3 IP Forwarding

Both IGP and EGP protocols for routing purposes are QC-aware. Routing protocol normally provide information for packet forwarding by taking into account the packet's associated l-QC. At the edge of autonomous domains, the *Traffic Conditioning & QC Enforcement* FB re-marks the packet's DSCP to an appropriate value with regard to the l-QC specified for the traffic stream.

The objective of the *IP Forwarding* function block is to maintain the Forwarding Information Base (FIB) of the router. A FIB stores all the routes, which have been selected/installed by the IP Routing processes that have been activated in the router and according to the results of each route calculation that has been launched by the activation of dynamic routing protocols. It is important to note that, in any case, the route selection is a decision which is made by Dynamic Intra/Inter-domain TE blocks. This decision is based on the qBGP information received, *pSLS* agreements, the QoS requirements that have been expressed by the appropriate parameters values in each SLS, that being processed by the MESCAL system.

It is possible for a router to keep many FIB tables, for example within the context of Meta-QoS-class deployment where there may be one FIB per Meta-QoS-class plane known by the router.

6.3.1 Interface Specification

Dynamic Inter-domain TE provides routing information for the egress ASBR for realising inter-domain TE paths while Dynamic Intra-domain TE provides routing information for the NEs within the domain for realising intra-domain TE paths. The interfaces of IP forwarding to these two function blocks are specified as below.

6.3.1.1 IP Forwarding Interface to Dynamic Inter- and Intra- domain TE

The following interface is used as the *IP Forwarding* interface to *Dynamic Inter-domain TE* for installing route-related information to ASBR type of NEs.

IP-FIB_CreateRouteEntry (Destination Prefix, DSCP, Next Hop Interface)

Creates a route entry where:

The *Destination Prefix* determines the destination to be reached using this route entry.

The *DSCP* is the DSCP value of the packets that use this route.

The *Next Hop Interface* determines the outgoing interface the packets will be forwarded for reaching the destination.

IP-FIB_DeleteRouteEntry (Destination Prefix, DSCP)

Deletes the route entry identified by both the *Destination Prefix* and *DSCP*.

IP-FIB_GetRouteEntryStats (Destination Prefix, DSCP)

Returns the number of bytes and the packets transmitted with regard to the route entry identified by *Destination Prefix, DSCP*.

IP-FIB_GetOutputInterfaceStats (Interface)

Returns the number of bytes and the packets received (and destined) by the output *Interface*.

Similar interface is used as *IP Forwarding* interface to *Dynamic Intra-domain TE* for installing route-related information to NEs within a network domain.

6.4 MPLS forwarding

The objective of the Data Plane block with respect to MPLS is to forward packets on LSPs. Dynamic Intra/Inter-domain TE must pass down the loose LSP path in order for head-end NE to request for the establishment of the LSP tunnels across the networks. The label distribution protocol should cross the boundary of domains for setting-up LSP end-to-end. The loose path used by a given tunnel at any point in time is already determined based on tunnel resource requirements and network resources such as bandwidth negotiated through *pSLSs* between domains. A packet crossing the MPLS-enabled network travels on a single tunnel that connects the ingress to the egress points across multiple domains.

Customer traffic at the LSP head-end should be classified in order to capture and reflect the negotiated *cSLSs*. The traffic profiles derived from the negotiated *cSLSs* should be enforced on the classified traffic before it actually enters the LSP. The *SLS Invocation Handling* function block should download appropriate traffic classification and conditioning configuration to ingress NE via the *Traffic Conditioning & QC Enforcement* function block. MPLS EXP field at the edge of the network can be set and change the ASBRs based on c/pSLS agreements.

7 MULTICAST

7.1 Multicast cSLS/pSLS

7.1.1 Introduction

Multicast cSLS/pSLS (mcSLS/mpSLS) are customer/provider service level specifications regarding multicast services with an ISP. Due to the inherent distinguishes in service model from unicast, the definition and specification of multicast cSLS/pSLS should also be different. The objectives of defining mcSLS/mpSLS are summarised as follows:

- From customer's viewpoint (mcSLS):
  - To allow for multicast end users (i.e., group members) express their individual QoS requirements in receiving multicast traffic from sources located in local/remote domains,
  - To set up an agreement on the maximum volume of multicast traffic each end user is allowed to receive. That means, the bandwidth consumption of each group that is subscribed by the customer should not exceed the upper limit that is specified in the mcSLS with the ISP. This implies that the bandwidth negotiation is based on per (S, G) group.
- From ISP's viewpoint (mpSLS):
  - To specify the QoS requirement (e.g., scope of domain level reachability for individual QoS classes) expressed to the upstream ISP in terms of multicast flows it is going to receive. Based on this type of mpSLS, the downstream ISP is able to set up mcSLS with its own multicast customers as well as to offer further mpSLS with other directly peering ISPs who request multicast transit services from it. Through this type of cascaded manner of mpSLS ordering/handling, QoS aware multicast service can be deployed globally,
  - To set up an agreement on the maximum volume of aggregated multicast traffic the requesting peer is allowed to receive. That means, the total bandwidth consumption of the multicast traffic by the requesting ISP should not exceed the upper limit that is specified in the SLS with its upstream peer.

Apart from the above objectives, the target mcSLS/mpSLS design should also take the some scalability considerations. Given a specific remote source S, multicast members attached to an ISP can express QC requirements on any group session provided by S. Therefore, the total number of mcSLS can be *n* times that of the corresponding unicast services between the two parties, where *n* is the number of groups rooted at S. This scalability issue should be taken into account especially when Source Specific Multicast (SSM) service model is adopted.

The detailed draft specifications of the algorithms have been suppressed in the public version of this document as they are in the process of being validated. The final versions will appear in D1.3.

7.2 Offline Multicast Traffic Engineering (OMTE)

7.2.1 Introduction

The objectives of Traffic Engineering (TE) may include: (1) efficient network dimensioning so that customer traffic demands can be satisfied while also keeping bandwidth consumption to a minimum; (2) control of traffic routes for achieving overall load balancing in the network. For multicast flows, bandwidth conservation is regarded as one of the most important tasks in traffic engineering, and this problem can be formulated into the directed Steiner tree problem, which is NP-Complete. If additional end-to-end QoS constraints and other TE objectives as load-balancing capability are embedded into the objective, the problem becomes even more complicated.

Despite the progress achieved for unicast services, traffic engineering for multicast services remains largely a dark area, especially in the IP layer. Recent research works have focused on Multi-Protocol Label Switching (MPLS) based online multicast traffic engineering, with the purpose of minimising multicast flow interference [KODIA03]. Scalability becomes an issue if MPLS explicit routing is adopted for multicast traffic engineering, given that a huge number of labels could be consumed for tree maintenance. Despite some research efforts on aggregating multicast traffic for reducing group states at the expense of extra bandwidth consumption [FEI01], mature solutions are still missing nowadays. On the other hand, pure IP, i.e. hop-by-hop routing approaches, present the following difficulties for multicast traffic engineering. First, the PIM-SM protocol uses the underlying unicast routing table for the construction of receiver-initiated multicast trees, and hence it is difficult to decouple multicast traffic engineering from its unicast counterpart. Bandwidth optimisation for multicast traffic can be formulated as the directed Steiner tree problem, which is NP-complete. The enforcement of Steiner trees can be achieved through packet encapsulation and explicit routing mechanisms such as MPLS. However, this approach lacks support from IP layer protocols, such as PIM-SM, due to RPF in the underlying multicast routing protocols. In PIM-SM, if multicast packets are not received on the shortest path with which unicast traffic is delivered back to the source, they are discarded for avoiding traffic loops. Given the inherently difference in shape between the shortest path tree used by PIM-SM and the optimised Steiner tree, the engineered multicast traffic for bandwidth optimisation through Steiner trees could result in RPF check failure.

The MESCAL solution will basically consider both IP and MPLS based multicast traffic engineering mechanisms including bandwidth consumption as well as load balancing. Also, the relevant task is decomposed into intra- and inter-domain parts. Since the proposed solution will be based on the existing routing protocols such as PIM-SM/MBGP, special considerations should be taken on RPF checking failure. Finally, given the fact that common network links are usually shared by both unicast and multicast traffic, it is desirable to provide a unified traffic engineering mechanism for the two type of services simultaneously, and this aspect will also be investigated in the project.

The detailed draft specifications of the algorithms have been suppressed in the public version of this document as they are in the process of being validated. The final versions will appear in D1.3.

7.2.2 Interface Specifications

Next_Hop_Update(OMTE to DMR)

The function parameters provided from OMTE to DMR are listed as follows:

- Source address prefix;
- Ingress router address for inter-domain group join;
- NEXT_HOP router address for intra-domain group join;
- Bandwidth availability on each intra- and inter-domain link;
- Reachability information on local and remote source prefixes.
- Configured QoS parameters such as delay, delay variation, jitter, loss probability etc.

7.2.3 Behavioural Specification

As we have mentioned, the basic task of OMTE is to optimise multicast traffic with QoS guarantees such as bandwidth and delay constraints that have been agreed in mSLSs. As far as intra-domain multicast services are concerned, traffic engineering also involves minimising overall bandwidth consumption within the network. Here we propose an IP layer TE approach for achieving this objective. Issues of inter-domain multicast TE will be included in our future research work. For intra-domain OMTE, in order to support dedicated mechanism for multicast traffic engineering, we base our approach on the Multi-topology ISIS (M-ISIS) routing protocol, which enables independent routing for unicast and multicast traffic.

7.2.3.1 M-ISIS based multicast TE

The conventional *OSPF* and *IS-IS* protocols only have a mono-viewpoint of the weight of each link in the network, and this influences path selections for both unicast and multicast traffic. In contrast, *M-ISIS* provides the original *IS-IS* protocol with the additional ability of viewing the weight of each link independently for different IP topologies. According to [PRZYG03], M-ISIS can support up to 128 different IP topologies. For multicast traffic, the Multi Topology identifier (*MT-ID*) of value 3 in *M-ISIS* is currently dedicated to the multicast *RPF* topology, i.e., the *RPF* table for *PIM-SM* can be populated using a set of independent link weights with *MT-ID* equal to 3. With this multi-topology capability of viewing link weights, it becomes possible that *PIM-SM* based multicast routing is completely decoupled from the underlying routing table for unicast traffic.

Figure 70 illustrates the basic framework of OMTE through optimised *M-ISIS* link weight setting. In a similar fashion to the unicast scenario, the network topology and the forecasted traffic demand from each multicast group are obtained as the input parameters for calculating the optimised link weights. After the link weights are computed through offline algorithms, they are configured in the network that runs the *M-ISIS* routing protocol with *MT-ID* equal to 3, which is dedicated to the multicast *RPF* table construction. Subsequently, each *M-ISIS* aware router computes shortest path trees according to this set of link weights and decides the NEXT_HOP router for a specific IP address/prefix. This type of NEXT_HOP information populates the multicast *RPF* table. When a *PIM-SM* join request is received, the router simply looks up the *RPF* table and finds the proper NEXT_HOP for forwarding the packet. In addition, the multicast forwarding information base (*FIB*) is dynamically updated for the incoming interface (*iif*) and outgoing interface (*oif*) list of each group.

Finally, it is worth mentioning that the M-ISIS link weight configuration can be performed on per QC basis, so that the NEXT_HOP to a specific address/prefix can be different in individual QC tree constructions. This makes it possible that PIM-SM join requests for different QCs are able to follow different paths when they are delivered towards the same source. Detailed description will be presented in the DMR section.

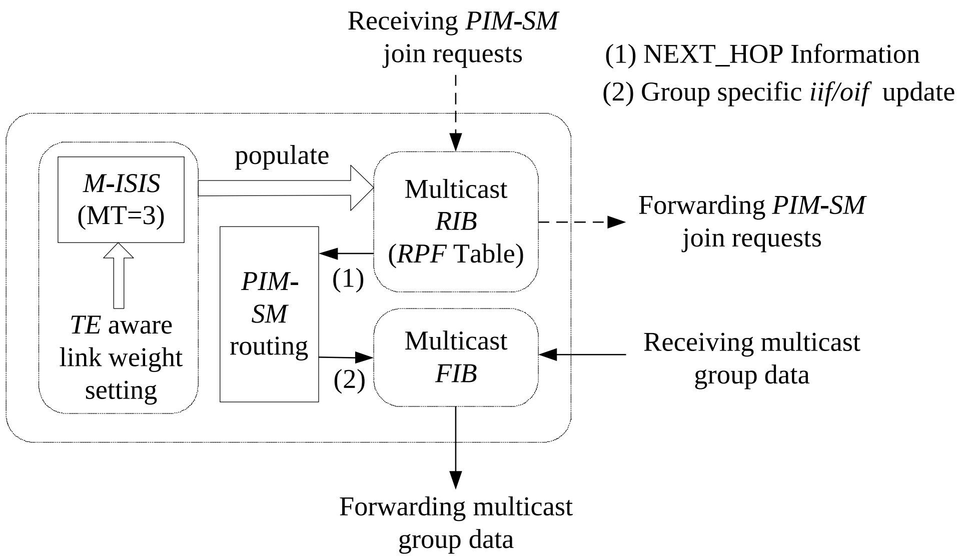

Receiving *PIM-SM* join requests
(1) NEXT_HOP Information
(2) Group specific *iif/oif* update
populate
M-ISIS (MT=3)
TE aware link weight setting
PIM-SM routing
Multicast *RIB* (*RPF* Table)
(1)
(2)
Multicast *FIB*
Forwarding *PIM-SM* join requests
Receiving multicast group data
Forwarding multicast group data

Figure 70. M-ISIS based multicast traffic engineering

7.3 Dynamic Group Management (DGM)

7.3.1 Introduction

The basic task of dynamic group management can be summarised into two parts: (1) to efficiently handle group membership dynamics with heterogeneous QoS requirements, and (2) to enforce admission control on multicast receivers to avoid network congestion due to overwhelming group subscriptions.

In the current best-effort based multicast services, Internet Group Management Protocol (IGMP) has the responsibility of managing multicast group dynamics. When a new group membership report is received, the Designated Router (DR) will trigger the PIM-SM protocol for sending the join request towards the source. In case that multicast group members demand different QoS requirements, IGMP should be extended to be QoS-aware such that the underlying routing protocol can be triggered to explore a proper path for the required QoS class. Moreover, special considerations should be taken when receivers with different QoS demands are attached to the common broadcast network. In this scenario, actions should be taken to prevent the receiver with lower QoS demand from accessing the same multicast data content but with higher QoS treatment that is requested by other members who might be charged more for his higher QoS services.

In the unicast scenario, admission control applies to external data sources during the SLS invocation period to prevent network congestion. When receiver-initiated multicast services are considered, sender oriented admission control is far from sufficient, since packets can be replicated anywhere due to multiple join requests from group members. This aspect explains why admission control should be performed at the receiver side when the multicast SLS is activated. In effect, network bandwidth can be over-reserved during the offline TE phase for efficient utilisation of resources, and this incurs possible congestion when overwhelming multicast SLSs are invoked simultaneously. If the DR receives a join request with the QoS requirement that the network resources cannot handle, this request should be rejected. To achieve this, the current IGMP should be extended such that excessive group membership reports are suppressed and the underlying routing protocol will not be triggered for join request delivery at the DR in time of congestion.

7.3.2 Interface Specification

Group_Member_Invocation(DGM to DMR)

When the join request from a new group member is notified in DGM, the following parameters should be passed to DMR:

- Source address;
- Group address;
- Specification of QoS requirements (e.g. QCs) by the new receiver.

7.3.3 Behavioural Specification

In our proposed *DGM* solution, each QoS class (QC) provided by the ISP is uniquely encoded into a class *D* address in the *SSM* address range 232/8 (see Table 12). In such a situation, the interpretation of the *SSM* address tuple (*S*, *G*) becomes straightforward: *S* identifies the address of the information source and *G* stands for the *QC* level (we name it *QoS* channel) that is available from *S*. The advantage of this scheme is that QoS requirement handling for individual receivers is translated into multicast group management, which can be directly fulfilled using IGMPv3 on a broadcast LAN. On the other hand, encoding QCs into *SSM* group address solves scalability problems in terms of QoS state maintenance at DiffServ core routers during dynamic multicast tree construction, and this will be specified in the *DMR* section. It should be noted that any class D address that does not belong to 232/8 is not considered to have such functionality. In effect, the maximum number of *QCs* in DiffServ is restricted by 6 bits of the *DSCP* field, and the allocation of 64 dedicated class *D* addresses will not cause any scalability problem in the usage of *SSM* address range that contains 2^{24} addresses. However, there is one restriction regarding this approach in implementation. Since the *QoS* channel is source specific, it is impossible for a single source with a unique *IP* address *S* to send multiple data streams with different contents. In the classic *SSM* model, an information source can be simultaneously involved in multiple groups because (*S*, *G1*) and (*S*, *G2*) are completely independent with each other. To solve this problem, the content provider may use multiple unicast addresses, each for a particular group/application.

SSM group address	*QoS Class*
G1 (232..*.*)*	*QC1*
G2(232..*.*)*	*QC2*
…	…
Gn(232..*.*)*	*QCn*

Table 12. SSM QC encoding table

The management of group join requests with heterogeneous QC demands is as follows. Once an end user wants to subscribe to a multicast service provided by the source *S* in a desired *QoS* channel (i.e., QC), it will send an *IGMPv3* (*S*, *G*) group membership report to its Designated Router (*DR*), where *G* is the associated group address mapped to that *QC*. On receiving this report, the *DR* will send an (*S*, *G*) join request towards *S* if this is the first (S, G) membership report appeared on the LAN. In case that multiple receivers subscribe to one multicast session provided by S but with different QCs, IGMPv3 should handle these membership reports independently since they contain different SSM group address. Finally, it is worth mentioning that admission control for multicast receivers is also on per (S, G) channel basis. That means, receiver-oriented mcSLS invocations for the same multicast source S but with different QC requirements are performed independently.

7.4 Dynamic Multicast Routing (DMR)

7.4.1 Introduction

Dynamic multicast routing refers to the procedure of multicast tree construction with dynamic group membership updates (i.e., mcSLS activation) with heterogeneous QoS class requirements. The basic task is (1) to build the deliver tree(s) that satisfies the QoS demand of all the attached receivers, and (2) dynamic path selection for bandwidth conservation and load balancing purposes. This functionality can be split into intra- and inter-domain parts, which respectively corresponds to the traffic engineering components in the offline blocks located in the management plane.

In the intra-domain scenario, the PIM-SM routing protocol constructs multicast trees based on the underlying unicast routing table. Traditionally this routing table for both unicast and multicast is populated by intra-domain unicast protocols such as OSPF and ISIS. To decouple multicast routing from the unicast realm, Multi-topology ISIS [PRZYG03] is extended from ISIS and it can provide dedicated routing decisions for PIM-SM tree construction. As a result, it is possible that the routing process can be dynamically managed specifically for multicast QoS demands and traffic engineering purposes. In case of statistical traffic fluctuations within each RPC, the adapted PIM-SM can make dynamic routing decisions to avoid further potential congestion.

At the inter-domain level, multi-protocol BGP (MBGP) [BATES00] is currently used for advertising multicast source reachability information which gives input to the inter-domain PIM-SM group join towards remote sources. Within each RPC, if there is significant domain-level topology or resource availability changing, the QoS-aware MBGP (qMBGP) is responsible for advertising updated reachability information (e.g., MP_REACH_NLRI and MP_UNREACH_NLRI) such that PIM-SM is able to dynamically decide the path for inter-domain join request delivery. The most challenging issues in this scenario include: (1) the extension of qMBGP from MBGP for the eQC-aware reachablity information of inter-domain multicast sources and (2) the corresponding online adjustment of multicast path selection across multiple domains.

Another important issue of DMR is the multicast tree construction with different QCs that serve heterogeneous QC requirements. In DiffServ networks, there are two strategies for building multicast trees that support heterogeneous receivers. First, a single tree exhibiting all QCs can be constructed for each group session, and branches with lower QCs can be directly grafted onto the part of the tree that has higher QC treatment. We name this strategy the hybrid tree approach. Another solution is to build a dedicated multicast tree for each QC, which means that *k* trees are needed for a particular group where *k* is the total number of QCs the ISP is providing within the network. We name this approach per QC trees. The difference between the two types of the trees is shown in Figure 71. In this figure we assume that QC(i) is higher priority than QC(j) if i< j. The advantage of per QC trees lies in its simplicity in implementation and management, while the hybrid tree has its virtue in bandwidth and group state conservation. The choice between the two strategies is one of the basic issues that the DMR block is going to address.

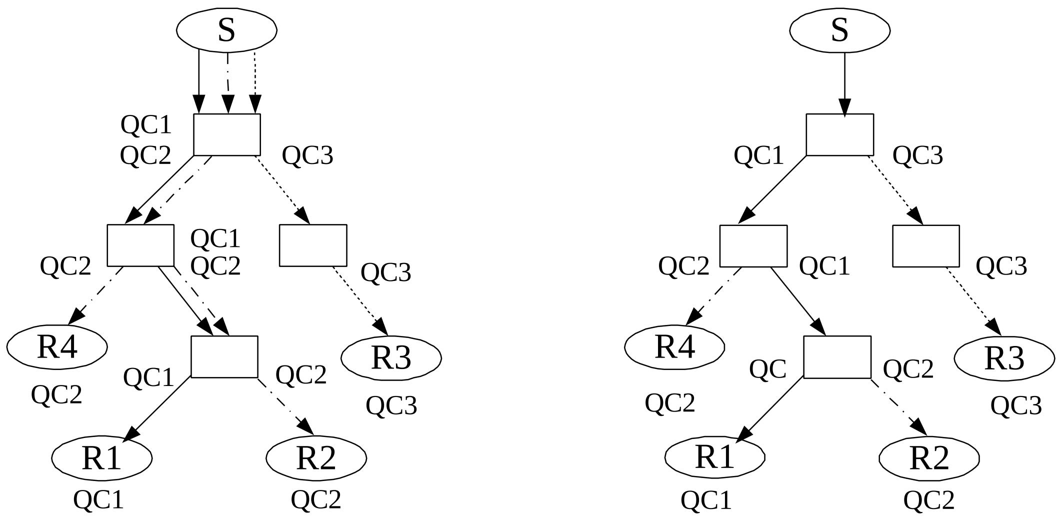


Figure 71. Per-QC tree vs. hybrid tree

7.4.2 Interface Specification

(1) Group_State_Update(DMR to MF)

This function basically informs the Multicast Forwarding block to create a new group state when a join request has received at a node for the first time. The MF will install the new (S, G) state as well as its iif and oif (i.e., the interface from which this join request has been received). The parameters include:

- Source address;
- Group address;
- Address of the iif;
- Address of the new oif.

(2) Oif_List_Update(DMR to MF)

This function basically informs MF to update the outgoing interface list (oif) when a new join request packet is received at a node that has already obtain the group state (i.e., on-tree router). Hence the MF will instruct the core router to forward multicast data packet on the new oif. The parameters include:

- Source address;
- Group address;
- Address of the new oif.

(3) PHB_State_Update(DMR to PE)

When a join request is received at each core router, not only the corresponding oif list should be updated, but also the associated PHB that is responsible for the multicast data packet treatment. The task of this function is to update at each oif the new PHB state with the dynamics of received group joins having different QC requirements. It should be noted that how this state is updated depends on the dynamic routing policy, i.e., whether per-PHB trees or a hybrid tree is adopted. The parameters include:

- Source address;
- Group address;

- Address of the new oif;
- The PHB used for requested QC;
- Routing policy ID (per QC tree vs. Hybrid tree).

(4) Iif_Update(DMR to RC)

Due to the network resource dynamics within each RPC, it is possible that the incoming interface (iif) for a particular source prefix is dynamically modified for the purpose of online bandwidth optimisation. This function provides RC the updated incoming interface for RPF checking. The parameters include:

- Source address (prefix);
- The new interface that is configured on the shortest path back to the source (prefix).

7.4.3 Behavioural Specification

We apply per QC trees in DMR. The most distinct advantage of this strategy is that bandwidth resources are much easier in provisioning for different QCs, because tree construction in one QC does not interact with any other QC in bandwidth consumption. From routing point of view, since M-ISIS can provide different routing table for multiple QCs, the PIM-SM join request for different QCs may follow different join paths towards the same source, thus resulting different tree shapes. According to [PRZYG03], M-ISIS can provide up to 128 different IP topologies, this means that the proposed scheme can support 128 QCs in maximum if M-ISIS is exclusively used for multicast services.

7.4.3.1 Intra-domain DMR

The construction of per QC trees is as follows. Once an end user wants to subscribe to the multicast service rooted at source *S* in a desired *QC*, it will send an *IGMPv3* (*S, G*) group membership report to its Designated Router (*DR*), where *G* is the associated group address mapped to that *QC*. On receiving this report, the *DR* will send a (*S, G*) join request towards *S* if the invocation has been admitted. This join request packet will either be intercepted by an on-tree router with the same (*S, G*) state or arrive at *S* itself.

In DMR, how to enable PIM-SM join requests with different group addresses (i.e., different QCs) to follow different join paths for achieving per QC tree constructions is a key issue. In the conventional SSM routing with best effort service, the underlying routing table is not group specific, but exclusively source specific. In this case, there should be additional mechanism needed for differentiating multiple routing topologies for different QCs. As we have mentioned in OMTE, M-ISIS is used to provide different routing tables for each QC. Hence, it is required that a mapping mechanism is used to link the group address carried in PIM-SM join packets to a specific M-ISIS routing table within a network. During the group join phase, when a router receives an (S, G) join request, it first finds the corresponding routing table by mapping the group address G into an M-ISIS MT-ID. Thereafter, the router looks up the routing table with that MT-ID and finds the NEXT_HOP for the source S. Finally, it forwards the (S, G) join packet on the interface associated with that specific NEXT_HOP. If there exist multiple NEXT_HOP entries leading towards the same source, the router should be allowed to deliver join requests in an ECMP style for load balancing purpose. From the above description, we notice that one extra mapping list should be maintained within each router such that group address can be linked to a specific QC topology identified by a dedicated MT-ID of the M-ISIS routing protocol.

Apart from the relationship between group address and MT-ID, considerations should also be taken on the interactions between group addresses and DSCPs, and this will be illustrated in the PHB enforcement section.

7.4.3.2 Inter-domain DMR

Inter-domain DMR is supported by Multi-protocol BGP (MBGP). Similar to the unicast scenario, inter-domain group joins are based on QoS aware MP_REACH_NLRI configuration at border routers within each AS. Within an RPC, binding selection for multicast services decides one or multiple MBGP edge routers for delivering inter-domain join requests towards a remote source address/prefix. During the mpSLS invocation period, group join packets can dynamically choose different edge routers decided by binding selection according to the instant QoS conditions and bandwidth consumption. As a result, the constructed inter-domain multicast tree can be dynamically adjusted due to the different join paths.

One of the challenges in handling inter-domain QoS delivery lies in the fact that ISPs have heterogeneous DiffServ configuration policies. For example, each DiffServ domain might have different number of QCs, and meanwhile the mapping between group address (i.e., QC identification in the PIM-SM join request) and the M-ISIS MT-ID is not necessarily consistent in all domains for the purpose of flexibility. This requires that the locally selected group address for QC identification should be made known to foreign domains, which is similar to the DSCP usage in unicast services. To achieve inter-domain per QC trees, we propose a swapping mechanism for group address at the border router of adjacent domains. Assume that a group join request needs to travel two adjacent domains A and B to reach the group source S, and it is required that the corresponding multicast flow should be treated with a particular eQC formed by lQC(A) and lQC(B) in the two domains respectively. In this case, the group address identifying lQC(A) in domain A should be changed into the address that corresponds to lQC(B) in its adjacent domain. This is because, for lQC(A), the ISP of domain A maps group address G(A) to MT-ID(A) for delivering the join packet within the local AS, but G(A) might not be recognised in domain B, or this address is mapped to some other lQCs using different routing tables. Hence, a swapping table should be agreed between two domains, so that each AS is able to perform correct local mapping between group address and MT-ID for the construction of inter-domain per QC trees.

7.5 PHB Enforcement (PE)

7.5.1 Introduction

Since multicast data packets can be replicated at core routers, additional issues arise for the corresponding PHB Enforcement. First, the join request from group members should be able to inform core routers in the tree about the associated QoS classes, so that the latter is able to enforce corresponding PHBs at the outgoing interfaces leading to heterogeneous receivers. Hence the first requirement of multicast PHB enforcement is to extend PIM-SM join request packet for inclusion of QoS class requirements that can be met with the configured PHBs. Second, as multicast trees are maintained through group states at core routers, if the outgoing interfaces of a router are associated with different QoS classes for the same (S, G) group, the maintained group state should also be extended for the associated QC information. We name this extra information at each outgoing interface *QC state*. It can be inferred that this type of state extension should take place at each outgoing interface of the (S, G) group.

The PHB Enforcement function block only provides some mechanism for supporting multicast services, and it does not output any input/output interface to other blocks.

7.5.2 Interface Specification

The PHB Enforcement interfaces are shown in Figure 73.

7.5.3 Behavioural Specification

SSM group address is used for carrying QC requirements from group members during the group join procedure. When multicast data packets are delivered backwards along each QC specific multicast tree, the PHB treatment is still based on Diffserv DSCP value. In our proposed scheme, QC states mentioned in section 1.6.1 are not needed to be maintained at core routers, because they can be directly reflected by (S, G) group states. This means that the issue of scalability at core routers is avoided. More detailed analysis on this can be found in [WANG04]. However, considerations should be taken on how the group address in a multicast packet is translated into a proper DSCP value for multicast data treatment. To solve the problem, we need a table at edge routers to map SSM group addresses (identifying QCs) into a specific DSCP value. The relationship of group addresses and DSCP values is somehow different from its relationship with MT-ID (see section 1.5.3.1). The most distinguished point is that: the mapping between group address and DSCP only takes place at *edge routers*, while each core router must know how to map one group address into a proper MT-ID in M-ISIS based PIM-SM routing.

For intra-domain multicast services, the edge router attached to the source is responsible for the translation of group address into a specific DSCP. When the edge router attached to a source receives a multicast data packet, it will mark the DSCP value of this packet coming from the source according to the pre-configured mapping table between group addresses and DSCPs, which should be agreed in the SLS between the source and the ISP. For example, when the edge router attached to source S receives an (S, G) multicast packet, the router first checks the locally maintained mapping table, and it then finds the DSCP value that is associated with group address G. Finally the edge router uses this value to mark the (S, G) multicast packet that will be injected into the network. When a core router that is already on the (S, G) tree receives another (S, G) join request from a new interface, it simply duplicates the multicast packet and forwards on the new oif. It should be noted that the DSCP value of the new packet is automatically inherited from the incoming packet, and this guarantees that each (S, G) tree is also a QC specific tree.

In the inter-domain scenario, group address swapping at border routers of adjacent domains is also needed for correct DSCP usage, with the reason being the same with MT-ID described in section 1.5.3.2. As it is shown in Figure 72, since group address swapping is performed at edge routers, inter-domain DSCP swapping is not necessary because the DSCP value to be used in the local domain can be obtained from the mapping of the swapped group address at the local edge router. We should also emphasise that the group address in both join packets and multicast data packet should both undergo such type of swapping in the inter-domain scenario. As a summery, group address swapping is double folded: first it enables inter-domain per QC tree construction by means of local mapping with MT-ID during the group join procedure(routing), and second, it enables correct PHB enforcement by means of local mapping with DSCP value (scheduling).

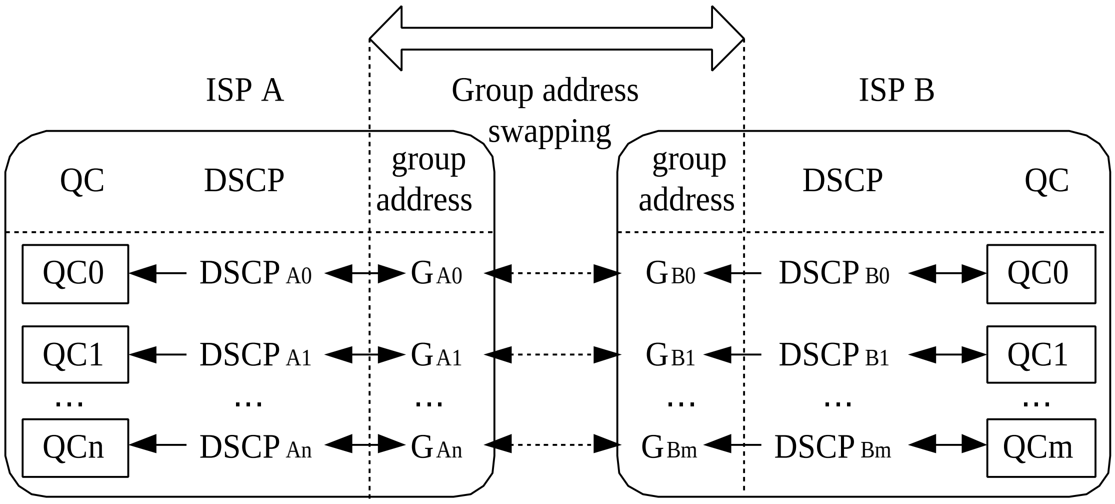


Figure 72. Inter-domain group address swapping

7.6 Multicast Forwarding (MF)

7.6.1 Introduction

The multicast forwarding part in the MESCAL project is not appended with additional functionalities compared to the conventional forwarding mechanism for multicast packets. On the other hand, the treatment of replicated packets with different PHBs is described in the PHB enforcement block.

7.6.2 Interface Specification

(1) PHB_Lookup(MF to PE)

During multicast data transmission, this function returns to MF how to schedule the data packets with proper PHBs at each outgoing interface. This action is based on the PHB state maintained at the PE block. The parameters include PHB state

(2) Iif_Lookup(MF to RC)

This function returns valid incoming interface to MF. If the data packet is not coming from this returned interface, it will be discarded. The parameters include:

- Source address/prefix;
- The address of the valid iif for the source (prefix).

7.6.3 Behavioural Specification

MESCAL will not study Multicast Forwarding any further.

7.7 RPF Checking (RC)

7.7.1 Introduction

RPF checking is a simple but efficient mechanism for preventing traffic loops during packet delivery in IP multicast and SSM. Since the MESCAL solution for multicast services will be based on the SSM model, RPF checking should be retained. On the other hand, there will be no extra adaptations on the RPF checking itself for QoS support.

MESCAL will not study RPF Checking any further.

7.8 Overall interface relationship

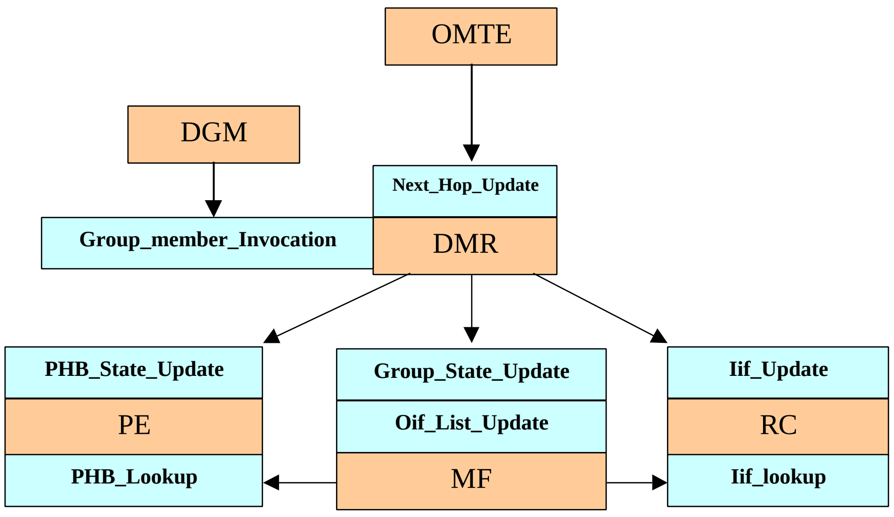


Figure 73. Overall multicast interface relationship

8 REFERENCES

[BATES00] T. Bates et al, “Multiprotocol Extensions to BGP4,” RFC 2858, June 2000.

[BELE02] S.Belenki “An Enforced Inter-Admission Delay Performance-Driven Connection Admission Control Algorithm,” *ACM SIGCOMM*, April 2002, Vol. 32, No. 2, pp. 31-41.

[BLAK98] S. Blake, et al, “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[BON01] T.Bonald, A.Proutiere and J.W.Roberts “Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding,” *IEEE INFOCOM 2001*, Vol. 2, pp.1104-1012.

[Buriol] Buriol, L.S., M.G.C. Resende, C.C. Ribeiro, and Mikkel, “A Memetic Algorithm for OSPF Routing,” *Proceedings 6th INFORMS Telecom*, pp. 187-188, 2002.

[CHAIT02] Y.Chait, C.V.Hollot, V.Misra, D.Towsley, H.Zhang and J.Lui “Providing Throughput Differentiation for TCP Flows Using Adaptive Two Colour Marking and Multi-Level AQM,” *IEEE Infocom* 2002, New York, NY, 23-27 Jun. 2002.

[CHARZ01] J.Charzinski “Problems of Elastic Traffic Admission Control in an HTTP Scenario”, *Proceedings of IWQoS* 2001.

[Cisco] Open Shortest Path First, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ospf.htm.

[D1.1] Paris Flegkas et al., “Specification of Business Models and a Functional Architecture for Inter-domain QoS Delivery”, MESCAL Deliverable D1.1, 19 June 2003.

[D1.4] H. Asgari et al., “Issues in MESCAL inter-domain delivery: technologies, bidirectionality, interoperability and financial settlements,” MESCAL Deliverable 1.4, 30 January 2004.

[DS] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[FEI01] Aiguo Fei et al, “Aggregated Multicast with Inter-Group Tree Sharing,” *Proceedings of Third International Workshop on Networked Group Communications (NGC2001),* UCL, London, UK, November 7-9, 2001.

[Fortz00] Fortz, B., and M. Thorup, “Internet traffic engineering by optimizing OSPF weights,” paper presented at INFOCOM 2000. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE, Vol.2, 2000.

[Fortz02a] Fortz, B., and M. Thorup, “Optimizing OSPF/IS-IS weights in a changing world,” *IEEE Journal Selected Areas in Communications*, May, 20, 756-767 2002.

[Fortz02b] Fortz, B., J. Rexford, and M. Thorup, “Traffic engineering with traditional IP routing protocols,” *IEEE Communications Magazine*, 40, 118-124 2002.

[FRE01] S.B.Fredj, S.Oueslati-Boulahia, and J.W.Roberts “Measurement-based Admission Control for Elastic Traffic,” in *Proceedings of ITC17*, Sept. 2001.

[GOD02a] Danny Goderis et al., “Service Level Specification Semantics, Parameters and Negotiation Requirements,” draft-tequila-diffserv-sls-02.txt, January 2002.

[GODER02b] D. Goderis et al, "A Scalable Service-Centric IP Quality of Service Architecture for Next Generation Networks", NOMS 2002.

[GUER91] R.Guerin, H.Ahmadi, and M.Naghshieh “Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, pp.968-98, September 1991.

[GUN92] R.Guerin, and L.Gun “A Unified Approach to Bandwidth Allocation and Access Control in Fast Packet-Switched Networks,” *IEEE INFOCOM* 1992, Vol. 1, pp. 1-12.

[HUST] G. Huston, “ISP Survival Guide,” John Wiley and Sons 1998. ISBN 201-3-45567-9

[IAB] Atkinson, R., Floyd, S., "IAB Concerns & Recommendations Regarding Internet Research & Evolution", draft-iab-research-funding-01.txt, July 2003.

[IDR] http://www.ietf.org/html.charters/idr-charter.html.

[JOVE01] L.Fabrega, T.Jove, A.Bueno, J.L.Marso “An Admission Control Approach for Elastic Flows in the Internet,” *9th IFIP Working Conference on Performance Modelling and Evaluation of ATM & IP Networks*, Budapest, June 2001.

[KODIA03] M. Kodialam et al, “Online Multicast Routing with Bandwidth Guarantees: A new Approach Using Multicast Network Flow,” *IEEE/ACM* Trans. on Networking, Vol. 11, No. 4, pp676-686, 2003.

[MAS99] L.Massoulié and J.W.Roberts “Arguments in Favour of Admission Control for TCP Flows,” In *proc. ITC 16*, Edinbourg, June 1999.

[MAY01] G.Iannaccone, M.May, and C.Diot “Aggregate Traffic Performance with Active Queue Management and Drop from Tail,” *Computer Communications Review*, July 2001.

[MESCAL] The IST MESCAL project; website: www.mescal.org.

[MYK03] E.Mykoniati, C.Charalampous, P.Georgatsos, T.Damilatis, D.Goderis, P.Trimintzios, G.Pavlou, and D.Griffin “Admission Control for Providing QoS in Diffserv IP Networks: The TEQUILA Approach,” *IEEE Communications Magazine*, January 2003, Vol. 41, No. 1, pp. 38-44.

[NIC99] K. Nichols, V. Jacobson, L. Zhang, “A Two-bit Differentiated Services Architecture for the Internet,” RFC 2638, July 1999.

[PRAT00] R.Mortier, I.Pratt, C.Clark, and S.Crosby “Implicit Admission Control,” *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 12, December 2000.

[PRZYG03] T. Przygienda et al, *“M-ISIS: Multi Topology (MT) Routing in IS-IS,”* draft-ietf-isis-wg-multi-topology-06.txt, March 2003 work in progress.

[QOSNLRI] Cristallo, G., Jacquenet, C, " Providing Quality of Service Indication by the BGP-4 Protocol: the QOS_NLRI attribute", <draft-jacquenet-qos-nlri-05.txt>, work in progress.

[RFC1771] Rekhter Y., Li T., "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC1997] Chandra, R., P. Traina, and T. Li, "BGP Communities Attribute", RFC 1997, August, 1996.

[RFC2328] Moy, J., “OSPF Version 2,” IETF Network Working Group RFC, 1998.

[RFC2434] Narten, T., Alvestrand, H., "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998.

[RFC2475] Blake, S., D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss (eds.), “An Architecture for Differentiated Services,” IETF RFC, 1998.

[RFC2702] Awduche, D.O., J. Malcom, J. Agogbua, M. O'Dell, and J. McManus (eds.), “Requirements for traffic engineering over MPLS,” IETF Network Working Group RFC, 1999.

[RFC3289] F. Baker, K. Chan, A. Smith, “Management Information Base for the Differentiated Services Architecture”, Standards Track RFC 3289, May 2002.

[RFC3290] Y. Bernet, S. Blake, D. Grossman, A. Smith, “An Informal Management Model for Diffserv Routers”, Informational RFC 3290, May 2002.

[RFC3392] Chandra, R., Scudder, J., “Capabilities Advertisement with BGP-4,” IETF RFC, November 2002.

[Riedl] Riedl, A., “A Hybrid Genetic Algorithm for Routing Optimisation in IP Networks Utilizing Bandwidth and Delay Metrics,” *Proceedings IEEE Workshop on IP Operations and Management (IPOM),* Dallas, USA, 2002.

[ROB98] J.W.Roberts and L.Massoulié. “Bandwidth Sharing and Admission Control for Elastic Traffic,” In *Proc. ITC Specialist Seminar*, Yokohama, October 1998.

[SGPT04] S.Georgoulas, P.Trimintzios and G.Pavlou “Joint Measurement- and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation Points,” to appear in *ICC 2004.*

[SHRO03] D.Eun and N.Shroff “A Measurement-Analytic Approach for Qos Estimation in a Network Based on the Dominant Time Scale,” *IEEE/ACM Transactions on Networking,* April 2003, Vol. 11, No. 2, pp. 222-235.

[TEQUILA] IST TEQUILA. www.ist-tequila.org.

[TEQUI,D1.4] P. Van Heuven et al., “D1.4: final architecture, protocol and algorithm specification,” TEQUILA, EU IST Project IST-1999-11253, 30 April 2002.

[TSE99] M.Grossglauser and D.Tse “A Framework for Robust Measurement-Based Admission Control,” *IEEE/ACM Transactions on Networking*, June 1999, Vol. 7, No. 3, pp.293-309.

[Walton] Walton, D., et al., "Advertisement of Multiple Paths in BGP", draft-walton-bgp-add-paths-01.txt, Work in Progress, November 2002.

[WANG04] N. Wang, G. Pavlou, “An Overlay Scheme for Provisioning Differentiate Services in Source Specific Multicast,” to appear in the *Journal of Computer Networks* (Elsevier).