



---

## D C1.2 – OAM Solution Specification for the Next Generation Carrier-Grade Ethernet Access Network

---

Wei Zhao, Jonathan Olsson  
Ericsson AB, Torshamnsgatan 2, S-164 80, Stockholm, Sweden  
[Wei.c.Zhao@ericsson.com](mailto:Wei.c.Zhao@ericsson.com), [Jonathan.Olsson@ericsson.com](mailto:Jonathan.Olsson@ericsson.com)

Yvette Kulik, István Moldován, Dorottya Vass  
Budapest Technical University, Magyar Tudósok Street, Budapest, 1117, Hungary  
[kulik@tmit.bme.hu](mailto:kulik@tmit.bme.hu), [moldovan@tmit.bme.hu](mailto:moldovan@tmit.bme.hu), [vass@tmit.bme.hu](mailto:vass@tmit.bme.hu)

István Gódor, Dávid Jocha  
Ericsson Hungary, Irinyi József u. 4.-20, 1117, Budapest, Hungary  
[Istvan.Godor@ericsson.com](mailto:Istvan.Godor@ericsson.com), [David.Jocha@ericsson.com](mailto:David.Jocha@ericsson.com)

Andreas Foglar  
Infineon Technologies D-81726 Muenchen  
[andreas.foglar@infineon.com](mailto:andreas.foglar@infineon.com)

Erland Sundberg  
TeliaSonera, Vitsandsgatan 9, Farsta, Sweden  
[Erland.sundberg@teliasonera.com](mailto:Erland.sundberg@teliasonera.com)

Identifier:	Deliverable DC1.2
Class:	Report
Version:	V16
Version Date:	12/12/2005
Distribution:	Public
Responsible Partner:	EABS
Filename:	D C1.2_V16

## DOCUMENT INFORMATION

<i>Project ref. No.</i>	IST-6thFP-507295
<i>Project acronym</i>	MUSE
<i>Project full title</i>	Multi-Service Access Everywhere
<i>Security (distribution level)</i>	Public
<i>Contractual delivery date</i>	2005-12-20
<i>Actual delivery date</i>	2005-12-12
<i>Deliverable number</i>	DC1.2
<i>Deliverable name</i>	OAM Solution Specification for the Next Generation Carrier-Grade Ethernet Access Networks
<i>Type</i>	Report
<i>Status &amp; version</i>	Released Report, V16
<i>Number of pages</i>	65
<i>WP / TF contributing</i>	WPC1
<i>WP / TF responsible</i>	EABS
<i>Main contributors</i>	EABS, EABH, BUTE, IFX, TS
<i>Editor(s)</i>	EABS, EABH, BUTE, IFX, TS
<i>EU Project Officer</i>	Pertti Jauhiainen
<i>Keywords</i>	Service Provisioning, Resource Management, QoS, Admission Control, Interfaces, Service Assurance, Bandwidth Brokering, SLA, SLS, Policy, Service Administration Function, Service Provisioning and Negotiation Function, Network Resource Administration Function, Implementation
<i>Abstract (for dissemination)</i>	This deliverable describes a solution proposal of network and service management for carrier-grade Ethernet access network to support triple-play services. It details important issues for service provisioning such as service definition, self-provisioning, resource management, network control and interface design. It also specifies core functionalities of the proposed architecture, divided into service delivery and service assurance.

## DOCUMENT HISTORY

Version	Date	Comments and actions	Status
01	09/06/2005	First draft	creation
02	14/06/2005	Second draft	Revised after meeting
03	15/06/2005	Third draft	Changes made by EABS
	27/06/2005	Group members claim chapters in ToC	
	28/06/2005	ToC prioritized by EABS	
	11/07/2005	Text added to chapter 3 by ETH and BUTE	
04	15/07/2005	3.2 elaborated by ETH	
05	02/08/2005	2.1 first draft done by EABS 3.1,3.2 further elaborated by ETH 3.3, 3.4 further elaborated by BUTE	
06	19/08/2005	2.2, 3.5 first draft done by EABS 3.1,3.2 further elaborated by EABH	
07	24/08/2005	1.1.2,1.2.2,1.2.3,3.6 first added by EABS 3.5 updated by EABS 4.1,4.4 first added by BUTE	
08	01/09/2005	3.1,3.2,4.2 modified by EABH References and figures edited by EABS 4.1,4.4 revised by BUTE 3.6 further elaborated by EABS	
09	15/09/2005	4.1.5 added by BUTE	

		<p>4.4-4.7 moved to a new chapter 5</p> <p>3.6 further elaborated by EABS</p>	
10	05/10/2005	<p>3.4,4.1.5 updated by BUTE</p> <p>4.3 added text by EABS</p>	
11	20/10/2005	<p>Comments added by EABH</p> <p>Editorial revision by EABS</p> <p>1.2.1 added by TS</p> <p>Changes made by EABS according to EABH's comments</p>	
12	03/11/2005	Modified by BUTE	
13	11/11/2005	<p>Added new text in chapter1 by IFX</p> <p>Chapter 6 finished by EABS</p> <p>Chapter 7 modified by BUTE and EABS</p> <p>Editorial work done by EABS</p>	
14	18/11/2005	<p>Chapter 4,7 updated by EABS</p> <p>Chapter 5 revised by BUTE</p> <p>More editorial work done by EABS</p>	
15	21/11/2005	<p>Chapter 3.3 modified by IFX</p> <p>Chapter 6.2.1, 3.1 modified by EABS</p>	
16	12/12/2005	<p>Chapter 1 and 2 swapped and content updated by EABS</p> <p>Editorial changes by EABS</p> <p>Chapter 3 commented by TS</p> <p>Chapter 3.3 updated by BUTE</p>	<p>Final Version before MUSE Quality Review</p>

## TABLE OF CONTENTS

DOCUMENT INFORMATION .....	2
DOCUMENT HISTORY .....	3
TABLE OF CONTENTS .....	5
LIST OF FIGURES AND TABLES .....	6
ABBREVIATIONS .....	7
REFERENCES .....	8
EXECUTIVE SUMMARY .....	10
<b>1 INTRODUCTION .....</b>	<b>11</b>
1.1 Scope of this document .....	11
1.2 Underlying Network Architecture .....	11
<b>2 BACKGROUND .....</b>	<b>12</b>
2.1 Relevant Work .....	12
2.1.1 MUSE DTF1.3 .....	12
2.1.2 CADENUS Model .....	14
2.1.3 TISPAN RACS .....	16
2.2 OAM functions in the data plane .....	17
2.2.1 ATM layer OAM functions .....	18
2.2.2 Lessons learned from ATM OAM .....	18
2.2.3 Ethernet OAM functions .....	19
2.2.4 IP layer OAM functions .....	19
2.2.5 Improvement of IP layer OAM functions .....	20
<b>3 OAM SYSTEM ARCHITECTURE .....</b>	<b>21</b>
3.1 Service Definition .....	22
3.2 Service Self-provisioning .....	24
3.2.1 Self-provisioning from the user's point of view .....	25
3.2.2 Self-provisioning from the network's point of view .....	26
3.3 Resource Manager .....	30
3.3.1 Admission control mechanisms .....	30
3.4 Network Controller .....	37
3.5 Databases .....	38
3.5.1 Global Service Registry .....	38
3.5.2 Service Depository .....	38
3.5.3 SLA/SLS Database .....	39
3.5.4 Policy Information Base .....	39
3.5.5 User Directory .....	40
3.6 Interface Specifications .....	40
3.6.1 Provisioning Interface .....	41
3.6.2 Service Negotiation Interface .....	41
3.6.3 Resource Negotiation Interface .....	41
3.6.4 Network Controller Interface .....	42
3.6.5 Device Controller Interface .....	42
3.6.6 Cross Management Domain Interface .....	42
<b>4 SERVICE DELIVERY .....</b>	<b>43</b>
4.1 Resource Management .....	43
4.1.1 Static management .....	43
4.1.2 Dynamic management .....	44
4.1.3 Bandwidth Brokering .....	44
4.1.4 Admission Control .....	45

4.1.5	Support of Absolute and Relative QoS.....	45
4.1.6	Device Control.....	46
4.1.7	N-RAF Mechanism.....	46
4.2	Service Subscription Management.....	48
4.2.1	SLA management.....	49
4.2.2	SLS Management.....	50
4.2.3	SLS and SLA Mapping.....	51
4.2.4	SLS and Policy Mapping.....	51
4.2.5	Detailed list of service subscription related service parameters.....	51
<b>5</b>	<b>SERVICE ASSURANCE FRAMEWORK.....</b>	<b>54</b>
5.1	Connectivity Fault Management.....	57
5.2	Performance Monitoring.....	59
5.2.1	Measurement objectives.....	59
5.2.2	The measurement process.....	60
5.2.3	Performance measurement functions.....	60
5.3	SLA verification.....	61
5.3.1	Measurement objectives.....	61
5.3.2	The measurement process.....	62
5.3.3	SLA verification functions.....	62
5.4	Root Cause Analysis.....	63
5.5	Advisor Module and Modification.....	64
5.6	Trend analysis.....	65
5.7	Adaptation and learning capabilities.....	65

## LIST OF FIGURES AND TABLES

FIGURE 1-1	ETHERNET REFERENCE MODEL FROM MUSE DA2.2.....	12
FIGURE 2-1	BASE MUSE MANAGEMENT ARCHITECTURE FROM DTF1.3.....	14
FIGURE 2-2	CADENUS MODEL [REF 33].....	15
FIGURE 2-3	TISPAN RACS.....	17
FIGURE 3-1	OAM SYSTEM ARCHITECTURE.....	21
FIGURE 3-2	WHOLE LIFE-CYCLE OF A SERVICE.....	22
FIGURE 3-3	SLOW LINK CAC.....	32
FIGURE 3-4	FAST LINK CAC.....	33
FIGURE 3-5:	EXAMPLE NETWORK WITH VLAN1 (GREEN).....	36
FIGURE 3-6:	EXAMPLE NETWORK WITH VLAN2 (BLUE).....	37
FIGURE 3-7	INTERFACES.....	40
FIGURE 4-1	PIPE DEFINITION.....	46
FIGURE 4-2	FLOW DIAGRAM BETWEEN THE ENTITIES.....	47
FIGURE 5-1	- SERVICE ASSURANCE FRAMEWORK.....	56
FIGURE 5-2	- RELATIONSHIP BETWEEN MEPS/MIPS AND PORT STATUS.....	58
TABLE 3-1	OVERVIEW OF THE MAIN PARAMETERS OF A SERVICE.....	23
TABLE 3-2	CONSTANT SERVICE PARAMETERS RELATED TO THE SERVICE DEFINITION.....	24
TABLE 3-3	QoS CLASS PARAMETERS.....	31
TABLE 3-4	MAXIMUM LOAD FOR FAST LINK.....	33
TABLE 3-5	MAXIMUM LOAD PER QoS CLASS.....	33
TABLE 4-1	CONSTANT SERVICE PARAMETERS RELATED TO SERVICE SUBSCRIPTION.....	52
TABLE 4-2	MODIFIABLE SERVICE PARAMETERS I.: TRAFFIC CHARACTERISTICS.....	53
TABLE 4-3	MODIFIABLE SERVICE PARAMETERS II.: THROUGHPUT CHARACTERISTICS.....	53

TABLE 4-4 MODIFIABLE SERVICE PARAMETERS III.: QUALITY OF SERVICE ..... 54  
 TABLE 5-1 AN EXAMPLE TABLE FOR GROUP CHECKING OF SLAS ..... 62

## ABBREVIATIONS

AIS	Alarm Indication Signal
AN	Access Node
ASP	Application Service Provider
BB	Bandwidth Broker
BER	Bit Error Rate
B2B	Business-to-Business
CAC	Call Admission Control
CC	Connectivity Check
CDV	Cell Delay Variation
CFM	Connectivity Fault Management
CNRM	Central Network Resource Manager
CPN	Customer Premise Network
DC	Device Controller
DCI	Device Controller Interface
DHCP	Dynamic Host Configuration Protocol
DSCP	DiffServ Code Point
EFM	Ethernet in the First Mile
EN	Edge Node
ENC	Event Notification Collector
eTOM	enhanced Telecom Operation Map
FAB	Service Fulfilment, Assurance and Billing
FCAPS	Fault, Configuration, Accounting, Performance and Security
FLR	Frame Loss Ratio
FM	Forward Monitoring
ISP	Internet Service Provider
LDAP	Lightweight Directory Access Protocol
MA	Maintenance Association
MEP	Maintenance End Point
MIB	Management Information Base
MIP	Maintenance Intermediate Point
NASS	Network Attachment Sub-system
NC	Network Controller
NCI	Network Controller Interface
NDP	Network Dependent Policy
NGN	Next Generation Network
NM	Network Management
N-RAF	Network Resource Administration Function
NSM	Network and Service Management
NSP	Network Service Provider
OAM	Operation and Maintenance
PM	Performance Monitoring
PI	Provisioning Interface
PIB	Policy Information Base

QoS	Quality of Service
RACS	Resource Admission Control Sub-system
RCA	Rout Cause Analysis
RDB	Resource Database
RDI	Remote Defect Indication
RFI	Remote Failure Indication
RNI	Resource Negotiation Interface
SA	Service Agent
SAS	Service Assurance
SAF	Service Administration Funtion
SLA	Service Level Agreement
SLAV	SLA Verification
SLS	Service Level Specification
SM	Service Management
SNI	Service Negotiation Interface
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SP	Service Provider
SPANF	Service Provisioning and Negotiation Function
S-VLAN	Service VLAN
TMN	Telecommunications Management Network
TOM	Telecom operation Map
VLAN	Virtual LAN
VoD	Video on Demand
VPN	Virtual Private Network
UDDI	Universal Description Discovery and Integration
VC	Virtual Connection
VP	Virtual Path
WSDL	Web Service Definition Language
WSF	Web Services Framework
XMDI	Cross Management Domain Interface
XML	Extensible Markup Language
XSD	XML Schema

## REFERENCES

Ref 1 MUSE Deliverable DTF1.3: OAM and NSM chain automation for multi-domain/multi-provider access networks, 14/01/2005

Ref 2 MUSE Deliverable DA1.1: Towards multi-service business models,30/06/2004

Ref 3 RFC3318 – Framework Policy Information Base

Ref 4 MUSE Deliverable DA2.2, Network Architecture and Functional Specifications for the multi-service Access and Edge, 12/2005

Ref 5 L. Kleinrock, Queueing Systems, Vol. I: Theory. Wiley, New York, 1975.

- 
- Ref 6 Hawkins, D Identification of outliers Chapman & Hall, London 1980
- Ref 7 Spiros Papadimitriou, Hiroyuki Kitawaga, Phillip B. Gibbons, Christos Faloutsos: Fast Outlier Detection Using the Local Correlation Integral; [http://www.intel-research.net/Publications/Pittsburgh/081620021325\\_99.pdf](http://www.intel-research.net/Publications/Pittsburgh/081620021325_99.pdf)
- Ref 8 Metro Ethernet Forum white paper: Metro Ethernet Networks - A Technical Overview
- Ref 9 Metro Ethernet Forum white paper: Metro Ethernet Services- A Technical Overview
- Ref 10 CISCO white paper: CISCO IOS IP Service Level Agreements
- Ref 11 CISCO white paper: Service Level Monitoring with Cisco IOS
- Ref 12 CISCO white paper: Accurate Network Performance Monitoring Using CISCO IOS IP Service Level Agreements
- Ref 13 MUSE Deliverable DA2.4 Network architecture and Function Specification for the Multi-Provider Access and Edge, 12/2005
- Ref 14 MUSE Deliverable DTF3.2 Definition of Test Suite and Minimum Requirements for Gateway Certification, 12/2005
- Ref 15 "Draft Standard for Local and Metropolitan Area Networks — Virtual Bridged Local Area Networks — Amendment 5: Connectivity Fault Management", IEEE 802.1ag, 2005
- Ref 16 "OAM Functions and Mechanisms for Ethernet based Networks", ITU-T Draft Recommendation Y.17ethoam, 2005
- Ref 17 "Web Services Architecture", W3C working group, February, 2004
- Ref 18 CADENUS: Creation and Deployment of End-user Services in Premium IP Networks
- Ref 19 "Relationships among ISDN, Internet Protocol and GII performance Recommendations", ITU-T Recommendation I.351, 10/2000
- Ref 20 "B-ISDN operation and maintenance principles and functions" ITU-T Recommendation I.610, 02/99
- Ref 21 "ATM protection switching", ITU-T Recommendation I.630, 02/99
- Ref 22 "Requirements for OAM functions in Ethernet-based networks and Ethernet services", ITU-T Recommendation Y.1730, 01/2004
- Ref 23 "Generic Requirements for Operations of ATM Network Elements", Bellcore GR-1248-CORE, Issue 4, November 1998
- Ref 24 "Operations, Administration, and Maintenance (OAM)", Clause 57 of draft Revision of IEEE Std 802.3-2002, 6th December 2004

---

Ref 25 "SNMP MIB definitions for Link Aggregation", Annex 30C (normative), draft Revision of IEEE Std 802.3-2002, 6th December 2004

Ref 26 "Requirements for support of Slow Protocols", Annex 43B (normative), draft Revision of IEEE Std 802.3-2002, 6th December 2004

Ref 27 RFC792, "Internet Control Message Protocol", J. Postel, 09/1981

Ref 28 RFC2463, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", Conta and Deering, 12/98

Ref 29 T.Cinkler, I.Moldován, A.Kern, Cs.Lukovszki, Gy.Sallai: Optimizing QoS Aware Ethernet Spanning Trees, MSAN 2005, Orlando, FL, USA, 2005 (p30-34)

Ref 30 T.Cinkler, I.Moldován, A.Kern: Optimized QoS Protection of Ethernet, DRCN 2005, Ichia, Italy, October 2005

Ref 31 IEEE 802.1s, Standard for local and metropolitan area networks: Amendment 3 to 802.1Q virtual bridged local area networks: Multiple Spanning Trees, 2001

Ref 32 TISPAN NGN Functional Architecture Release 1, Draft ETSI.ES.2xx.xxx.v<1.1.4>, February, 2005

Ref 33 G. Cortese, R. Flutem, P. Cremonese, E.P.F.L. Lausanne, S. D'Antonio, M. Esposito, S.P. Romano, A. Diaconescu, "CADENUS: Creation and End-User Services in Premium IP Networks", IEEE Communications Magazine, January 2003.

## EXECUTIVE SUMMARY

The OAM aspects covered by this report include both network management and service management. The solution proposed in this report stresses the service provisioning phase of the whole service life circle, including service definition, service subscription, self-provisioning, and resource management. The network management tasks are also touched as much as possible. The solution proposed in the document uses as a base the CADENUS [Ref 18] architecture model and extends it with more detailed description of each functional block as well as interface specifications. The standardization work at TISPAN, especially the RACS specification is taken into consideration especially for interface design in order for compatibility and future fixed mobile convergence.

### What's New in this document

The OAM solution is based on CADENUS model, but it has extended it extensively with much more detailed description inside each functional block.

Three new functional blocks have been defined:

- SPANF (Service Provisioning and Negotiating Function) corresponds to the Access Mediator in the Cadenius model.
- SAF (Service Administration Function) corresponds to the Service Mediator

- N-RAF (Network Resource Administration Function) corresponds to the Resource Mediator)

We have also defined interface specifications among different components, which is not included in the CADENUS model. The XMDI interfaces among different N-RAFs are proposed for end-to-end resource mediation, which is a new idea that may lead to further investigation in the future.

The document also proposes a novel way of using existing VLAN tag to support absolute and relative QoS at the same time.

The QoS provisioning model with the usage of a Central Network Resource Manager is not new. However, the adaptation to the Ethernet environment with VLAN based un-provisioned pipe model is a novel approach.

The Service Assurance framework, adapted to the Public Ethernet environment based on the draft standards of Connectivity Fault Management (802.1ag and [Ref 16]) and Performance Management ([Ref 16]) and other OAM mechanisms is new. The draft standards cover the basic OAM mechanisms for network management only, but no framework is specified for automatic service assurance. Furthermore, the **active** Root Cause Analysis method based on Petri nets and the learning alarm processing methods are also new.

## 1 INTRODUCTION

### 1.1 Scope of this document

This report is a deliverable within WPC1.1 and is aiming at providing an OAM solution proposal for Ethernet based broadband access network. The reference network architecture used in this document is the detailed architecture description used by the WPA2 architecture group, although the solution itself should not be limited only to the reference network architecture.

The OAM aspects covered by this report include both network management and service management. The solution stresses the service provisioning phase of the whole service life circle, including service definition, service subscription, self-provisioning, and resource management. The network management tasks are also touched as much as possible.

The document is organized as follows: after background and introduction, the OAM architecture is described, where important issues for service provisioning, such as service definition, service self-provisioning, resource management, network control, databases and interfaces are investigated and detailed. The follow-up chapters describe core functionalities of the proposed architecture, which is organized into service delivery and service assurance. In service delivery chapter, functions are organized in terms of a service delivery life cycle, which includes service subscription management and resource management. In service assurance chapter, different aspects of service assurance such as connectivity fault management, performance monitoring and SLA verification are investigated and a proposal about how these models should work together is described.

The expected outcome of this report will give a detailed description of OAM system architecture and functional blocks.

### 1.2 Underlying Network Architecture

The network reference model for Ethernet based access network shown in Figure 1-1 is described in [Ref 4].

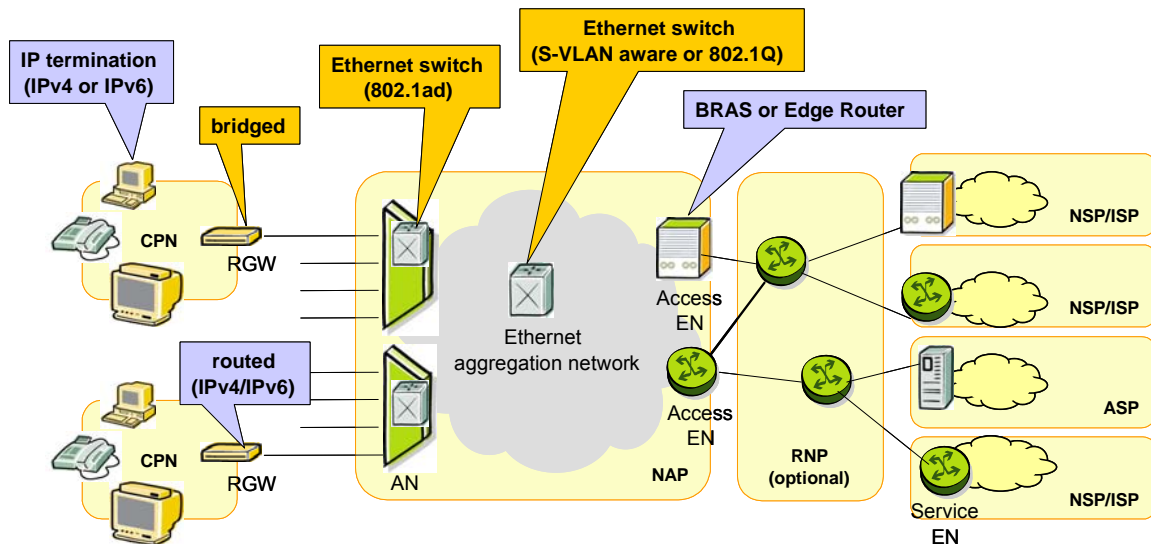


Figure 1-1 Ethernet Reference Model from MUSE DA2.2

There are several highlights of the access network architecture based on this reference model, which affect how the OAM system will be built:

- Ethernet should be used between CPNs and Edge Nodes.
- Intermediate Ethernet switches work in a point-to-point manner.
- Connectivity between end-users and the service providers (ASP, ISP, NSP) is provided by means of service bindings.
- Service binding is defined as static or semi-static with a number of attributes associated to it.
- Service binding attributes, such as QoS, Security, are dynamic.
- Service bindings are asymmetric meaning that upstream and downstream service bindings can have different properties.
- Service bindings in the Public Ethernet solution are created on L2, i.e. the MAC layer.

## 2 BACKGROUND

### 2.1 Relevant Work

#### 2.1.1 MUSE DTF1.3

The work in TF1.3 is addressing OAM and NSM chain automation on multi-domain/multi-provider Access Network. The deliverable DTF1.3 [Ref 1] presents and compares the two management reference models, ITU-T TMN model and TMForum eTOM model. The TMN model is built on the requirements to manage network equipment and networks as a bottom-up approach, whereas eTOM model is more a top-down approach to describe the enterprise processes required by a service provider.

---

Since the object of the MUSE project is to define an OAM system to configure the network from a service point of view, the DTF1.3 is describing in more detail the requirement for the three main eTOM processes, namely fulfilment, assurance and billing. There is also a section related to Security Management which describes a management of security concept intended to mechanize the application of various security and security management tools.

In the proposed management architecture two management domains has been defined, the Vendor management domain and the Operator management domain and in between a Mediation layer (see Figure 2-1).

The Vendor management Domain corresponds to a set of management functions directly implemented in network elements or in management systems provided by the network vendor. All managers located in this area are specific managers dedicated to each network vendor. The Operator management Domain corresponds to a set of applications allowing performing the three main eTOM processes: namely fulfillment, assurance and billing processes. The main feature of this domain is to be common to all services and all vendors. The Mediation Layer is considered as a part of Operator domain in order to adapt protocols and data to IS (Information System) requirements.

References from both eTOM (TMF) and TMN (ITU) models are used :

- Horizontal splitting is based on TMN layers:
- Vertical splitting is based on eTOM grouping process (Fulfilment, Assurance and Billing). They are mainly located in Operator Domain.

Depending on type of interface between Vendor and Operator domains and on the location of the Management system, four different solutions for the management are considered and compared in the work of TF1.3. The solution 4 is chosen as the preferred base for management architecture in the MUSE.

The Solution 4 has two management chains (OAM and IS) which are interconnected between themselves, through interfaces between the vendor Management Systems and the operator domain:

- one chain dedicated to OAM functions (Vendor Domain)
- another chain dedicated to IS operations (Operator Domain)

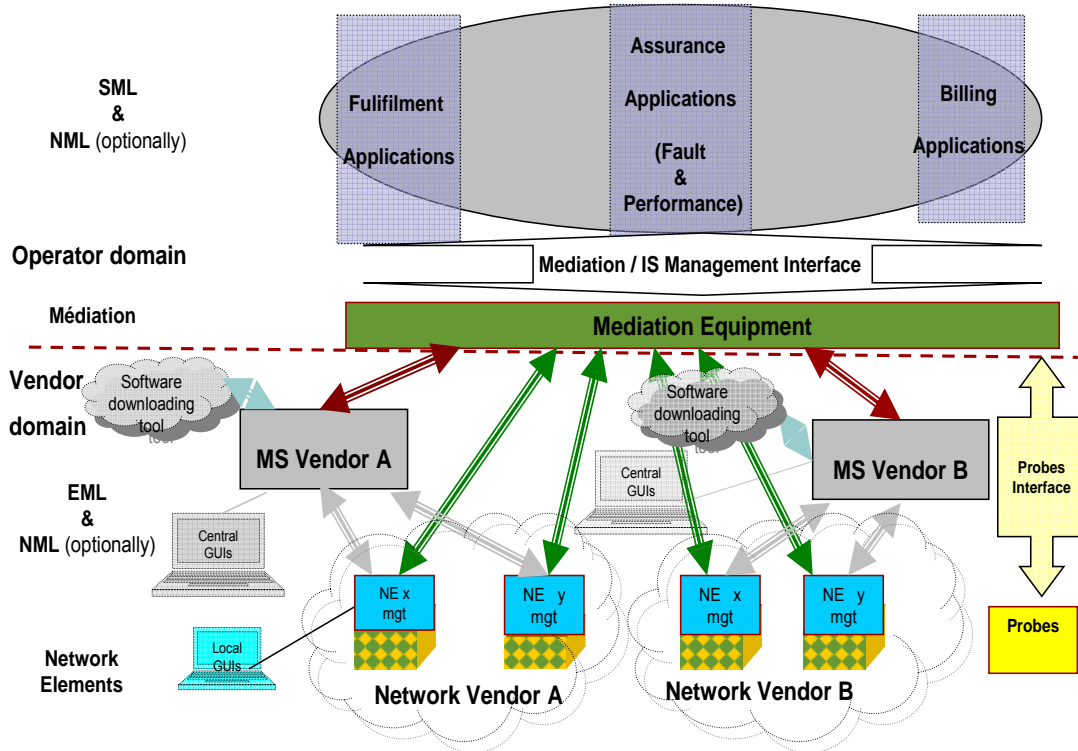


Figure 2-1 Base Muse management architecture from DTF1.3

The complexity of Solution 4 is that in addition to Southbound interfaces (towards NEs), MS must provide Northbound interfaces towards Operator Domain. The Mediation Level will be connected to numerous management interfaces (both from MS and each NE). Adaptation to IS requirements of protocols and data is needed, except if open interfaces are used. Also two types of management interfaces are required: towards Management System (rather vendor-specific ones) and towards Mediation Level (rather standard ones).

The architecture proposed in DC1.2 is in line with the solution 4 architecture from DTF1.3.

### 2.1.2 CADENUS Model

The aim of the Cadenus [Ref 18] project was to bring theoretical and practical contributions to the area of dynamic creation, configuration and delivery of services with QoS guarantees via the automated management of service level agreements, by defining a framework for the provisioning of advanced communication services in premium IP networks. Such networks might be characterized by a high degree of complexity, in terms not only of scale, but also of number of operators and technological heterogeneity. An innovative approach was taken to framework design, based on the concept of mediation. The CADENUS policy frame is depicted in Figure 2-2:

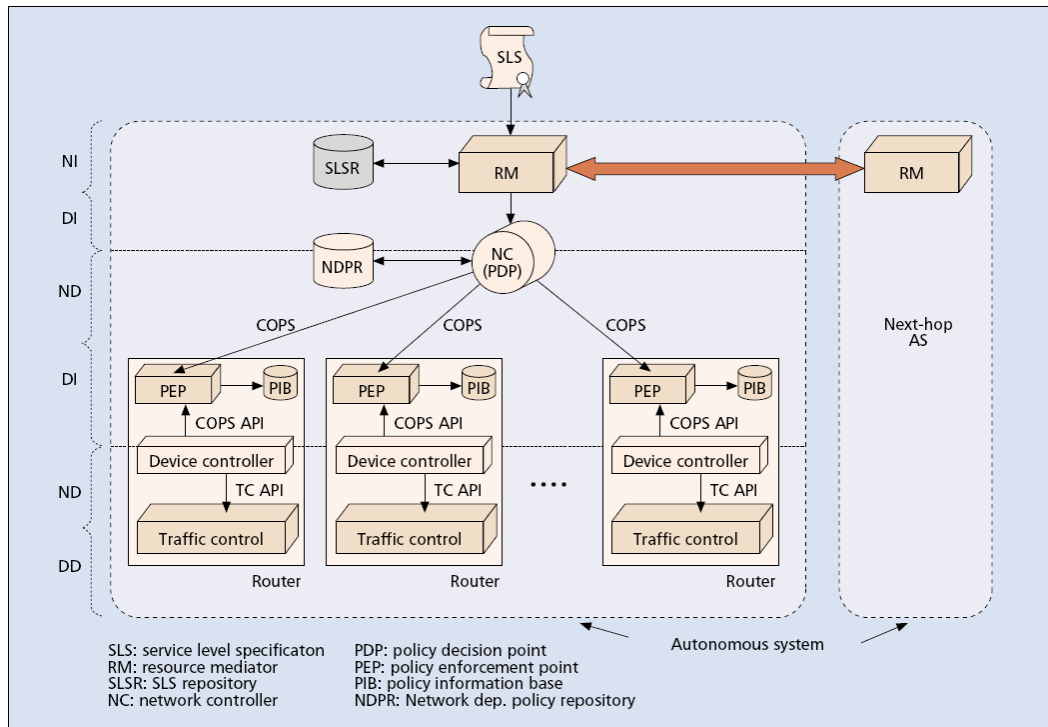


Figure 2-2 CADENUS Model [Ref 33]

Three major mediator components have been identified as needed to supervise the dynamic service creation and configuration process:

- Access mediator
- Service mediator
- Resource mediator

The access mediator is the entity that allows users to input their requests to the system (i.e., mediates the users' access to the system). The main responsibility of the access mediator is to present the user with a set of available services and corresponding service descriptions (e.g., type, cost) and allow the user to select, customize, and purchase these services via a harmonized interface easing the service selection process.

---

The access mediator may form associations with one or more service mediators to which requests are issued. Generally offline, the service mediator will take care of the creation of new services and their presentation in the service directory. It is the task of the service mediator to map the SLA from the access mediator into the associated SLS(s) to be instantiated in cooperation with the resource mediator(s). The SLA can therefore be seen as the interface between the access mediator and the service mediator.

The interface between the service mediator and the resource mediator is the SLS, ensuring independence from both the high-level view of a service and the specific network architecture employed. In a few words, a resource mediator can offer all that can fit inside an SLS.

The resource mediator may enforce SLS by means of a policy-based approach, which ensures correct operation of the network in a flexible and dynamic fashion. Policy-based management enables network administrators to shift from individual device management toward controlling a network as a whole: it allows network elements from different vendors, equipped with different capabilities, to be consistently configured.

By defining these three types of mediators, two strategic goals are achieved. First, the requirements of the emerging business models are met by addressing the various interaction types between users and providers, as well as between different providers (e.g., negotiations, service selection, profiling). Second, not only the service from the resource control and management, but also the service from the service creation machinery is separated. The former is a well-recognized requirement for next-generation networks, while the latter architectural feature opens standardization potential for service creation work. The three mediation components, together with the customer, are the entities taking part in the operational phase of the premium IP (PIP) services.

Solution proposal from DC1.2 will extend the CADENUS model by giving a much more detailed description of each mediator and also defining interfaces between different mediators.

### **2.1.3 TISPAN RACS**

TISPAN Release 1[Ref 32] is specifying an architecture for fixed broadband access network of NGN, which among other things specifies the resource admission control sub-system (RACS) needed for the access network. RACS belongs to the TISPAN transport control together with the Network Attachment Subsystem (NASS), which is shown in Figure 2-3.

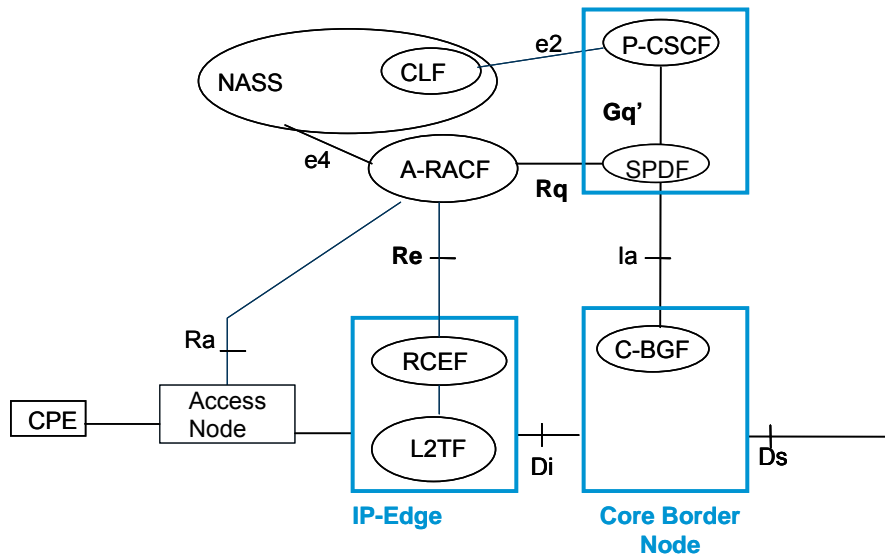


Figure 2-3 TISPAN RACS

TISPAN RACS has the following features:

RACS provides the support of policy-based framework. This means it provides a layered structure and a separation of services and network. RACS itself does not specify detailed mechanism and requirement for any specific access network, although it uses as an example the xDSL network.

RACS is session aware but service agnostic. It is designed mainly to provide QoS support on a per session basis. However, it does not exclude the possibility of supporting QoS on a per flow basis, which gives an anchor point to connect to 3GPP.

RACS provides to applications a mechanism to request and reserve resources from access network. To achieve this, however, the applications must be capable of triggering the QoS resource reservation, admission control and policy control capabilities of the network, which implies that proper interfaces must be defined between application layer and the network layer of the multi-service bearer architecture.

TISPAN RACS is meant to work together with 3GPP, aiming at providing solutions for fixed mobile convergence. The solution designed in DC1,2 keeps TISPAN RACS as a close reference especially for interface design, and this will make DC1.2 compatible for fixed mobile convergence scenario.

## 2.2 OAM functions in the data plane

Functions within the data plane can be divided in two groups, alarm functions to detect unpredictable events/errors and monitoring functions to supervise the performance of connections. We will use the more elaborated ATM layer OAM as benchmark (and to make benefit from lesson learned) for development of requirements for the Ethernet and IP network.

## 2.2.1 ATM layer OAM functions

ATM provides OAM layer functions described in ITU-T Recommendation I.610 [Ref 20] **Error! Reference source not found.** In addition I.630 [Ref 21] describes protection switching.

**Alarm functions** – this is the AIS/RDI mechanism known from SDH/SONET. It is triggered by failures in the lower (physical) layer or by ATM layer (e.g. misrouting) failure. In case of a physical layer failure Alarm Indication Signal (AIS) cells are inserted in forward direction in 1s intervals for all affected VP/VC. At the endpoints the AIS cells are looped back as Remote Defect Indication (RDI) cells. An escalation mechanism leads from VP level AIS to VC level AIS at VP endpoints. So within 1 second all affected receivers and senders are informed.

To detect ATM layer failures Continuity Check (CC) cells are inserted in 1s intervals at the origin of VC/VP. If at the endpoint neither CC nor user cells are detected for a certain time a failure is assumed and the AIS/RDI mechanism is triggered.

Errors are classified in 3 levels, anomaly, defect, failure. Anomalies are single events which can be repaired locally. A failure is treated locally by triggering the AIS/RDI mechanism in the data plane. Only if a defect persists for some time a failure is declared and signalled to the management plane.

**Monitoring functions** – there are two mechanisms, Loopback (LB) and Performance Monitoring (PM). LB cells have precise scope, such as end-to-end, segment or node-by-node for precise failure localisation.

PM is a complex function requiring special hardware. At the entry point Forward Monitoring (FM) OAM cells are inserted between blocks of user cells. They carry cell count, checksum and optional timing information. At the destination point the cell stream is evaluated and compared to the information carried in the FM cells. Precise information on how to evaluate and categorise the measured raw data in erroneous seconds, severely erroneous seconds etc. is given in Bellcore TR-1248 [Ref 23]. Due to the complexity of the PM function it was never deployed in large scale.

Disadvantage of PM is that only VC/VP with user cell streams can be measured. It is not possible to check a VC/VP before putting it into service. For this purpose the Testing function was foreseen for ATM, but rejected by standardisation. The idea was to generate cell streams with pseudo-random payload to measure unused VC/VP, for example before putting them into service.

**Other functions:** ATM OAM includes activation and de-activation of CC and PM functions via the data plane. Using respective OAM cells for example a node at one end of a VC/VP can initiate CC or PM function at the other end.

## 2.2.2 Lessons learned from ATM OAM

What was good in ATM:

- AIS/RDI mechanism extended from SDH/SONET
- CC mechanism to detect misrouting etc. even in the absence of user cells.
- Loopback mechanism with widely selectable scope for precise failure localisation
- Error escalation “anomaly – defect – failure” with filtering to prevent message flooding
- Well-defined data collection for PM data [Ref 23].

What was not so good in ATM:

- Complex PM mechanism which requires special hardware, CPU time, large amounts of data to be stored and additional data plane overhead for FM cells.
- No solution to measure cell jitter. The optional time field in Loopback or FM cells was not standardised.
- No solution to measure VC/VP in the absence of user cells. Testing function was not standardised.

### 2.2.3 Ethernet OAM functions

Ethernet OAM has been studied by EFM group as a mandatory function for carrier-grade Ethernet. The outcome was link level OAM which has been included in 802.3 as optional feature in clause 57 [Ref 24].

Link level means that the scope of the Ethernet OAM packets extends only over a collision domain or over a first mile link. In accordance with the broadcast nature of Ethernet the OAM frames are multicast frames. They are coded as a subtype of the so-called “slow protocol frames” specified in Annex 43B [Ref 26]. Slow means that not more than 10 slow protocol frames may be sent per second. The immediate consequence is that PM can not be supported. Also Ethernet OAM frames are VLAN agnostic. It is not possible to check the availability of a special VLAN.

The functions provided by Clause 57 OAM [Ref 24] are a subset of ITU-T Y.1730 [Ref 22].

**Alarm functions** – these are the Remote Failure Indication RFI (analogous to RDI in ATM) and Link Monitoring mechanism, both to signal unpredictable events. RFI indicates to peer(s) that the receive path of the local data terminal equipment is non-operational.

**Monitoring functions** – a LoopBack function is provided and a mechanism to poll any variable in the Clause 30 MIB [Ref 25].

**Other functions:**

- Activation of OAM is optional.
- A mechanism is provided that performs OAM capability discovery.
- A peer may be active or passive.
- An extension mechanism is provided and made available for higher layer management.

### 2.2.4 IP layer OAM functions

Management on the IP layer is specified in the Internet Control Message Protocol ICMP. It is described in RFC792 [Ref 27] for IPv4 and RFC2463 [Ref 28] for IPv6. ICMP has never been denoted as OAM function description, but the functionality of the messages is analogous to previously described OAM functions. Therefore in the following description of ICMP messages they are grouped in alarm and monitoring functions.

RFC792 specifies ICMP messages for IPv4.

**Alarm functions**

- Destination Unreachable (destination address unknown)
- Parameter Problem (invalid header parameter, e.g. invalid payload)
- Source Quench (backpressure in case of overload)

- Time Exceeded (time to live field zero)
- Redirect (shorter path possible)

#### **Monitoring functions**

- Echo (availability of a host/ address)
- Echo Reply
- Timestamp (measure round trip delay)
- Timestamp Reply
- Information Request (retrieve network address)
- Information Reply

The monitoring functions are grouped in request/reply pairs. Echo/echo reply messages are used by the PING function (see RFC2463), which is very useful, but today also is a major tool for hackers.

RFC2463 specifies ICMP messages for IPv6. RFC2463 groups them in error messages and informational messages. Hence the mapping to our nomenclature is obvious.

#### **Alarm functions** (ICMPv6 Error Messages)

- Destination Unreachable (destination address unknown)
- Parameter Problem Message
- Packet Too Big (no fragmentation in IPv6)
- Time Exceeded

#### **Monitoring functions** (ICMPv6 Informational Messages)

- Echo Request (availability of a host/ address)
- Echo Reply Message

### **2.2.5 Improvement of IP layer OAM functions**

Applying the principles of ATM OAM considering the lessons learned to IP layer ICMP messages and combining with useful features from Ethernet OAM could lead to a mature and powerful IP layer OAM functionality. Especially for IPv6 these new functions could be implemented as IPv6 networks are still much less deployed than IPv4 networks. Some examples are given here.

1. ICMP messages have only one reach scope, which is “to end”. Introducing the variable reach scope of ATM layer OAM cells would lead to “to gateway” Echo which is reflected by the residential gateway or “network-internal” Echo which is reflected at network boundary or to “per-node” Echo which is reflected by each node. These messages could be used for failure localisation or also to check connectivity up to the network border.
2. With the Timestamp message a definition of absolute time is available with a granularity of 1ms (32bit starting at midnight UT). It could be a tool for delay and jitter measurements using 1-point CDV and 2-point CDV mechanisms described in ITU-T I.351 [Ref 19].
3. CC mechanism combined with timestamp mechanism could be extended to IP layer to assure availability of service bindings with specified QoS.
4. The error escalation strategy of ATM (anomaly – defect – failure) could be useful for IP layer OAM as well.

### 3 OAM SYSTEM ARCHITECTURE

The SPC OAM architecture is an extension to the CADENUS architecture model, which is composed of following functional blocks shown in Figure 3-1.

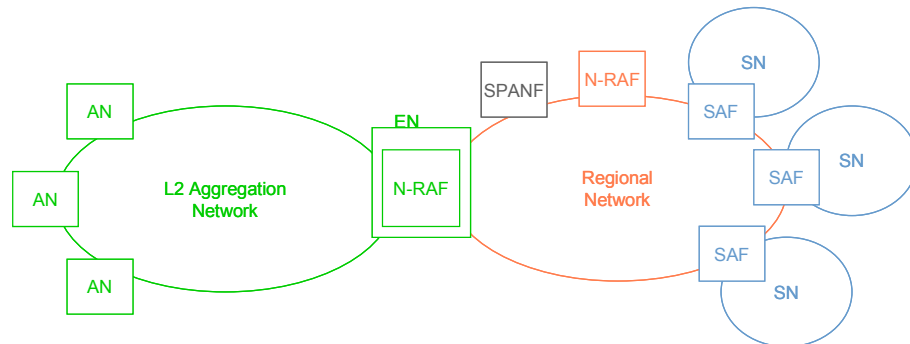


Figure 3-1 OAM system architecture

SPANF(Service Provisioning and Negotiation Function) is the connecting point of end users and service providers. It is an extension to the Access Mediator of the CADENUS model. It provides an interface towards different service providers through which service providers can publish their services to end users. It also provides a service portal towards end users where they can subscribe a service, modify their existing service subscription, select and activate services, and stop a service. The SPANF is also responsible for end user identity management which can be used for single sign-on or service usage authentication issued by service providers.

SAF(Service Administration Function) is managed and maintained by individual service providers for their own services. It is an extension of the Service Mediator of the CADENUS model. It is responsible to manage the whole service life cycle including service creation, publishing, charging, monitoring, AAA, repairing, service ending, etc. SAF has an interface towards SPANF where it publishes its mature services and fetch necessary user identity information for service authentication. It also maintains an interface towards network aware functionalities for service resource management.

N-RAF(Network Resource Administration Function) is responsible for all issues related to network resources. It is an extension of the Resource Mediator in the CADENUS model. It takes service resource request from SAF and changes it into network resource request, it then communicates with neighbouring N-RAFs in a distributed manner to check resource availability end-to-end. Well-known functionalities in N-RAF include bandwidth brokering, admission control, network topology management, etc.

In the following sections, the OAM architecture is introduced through the “whole life-cycle” of a service (not all parts are covered by MUSE or this document) including network administration functionalities behind the provisioning of a service (see Figure 3-2).

The “whole life-cycle” of a service starts with a business survey (which is out of the MUSE’s scope and neither further discussed nor analyzed in this document). The first step in this work (Step 1 in Figure 3-2), the *Service Definition* in section 3.1, is described by the introduction of the most important parameters of a service and the way of defining them in different places in the supply chain (Step 1 in Figure 3-2). The *Service Discovery* (Step 2 in Figure 3-2), the *SLA & SLS negotiation* and the input for *Policy Creation* (Step 3 & 4 in Figure 3-2) and the *Service Binding Setup, Activation & Deactivation* (Step 6 in Figure 3-2) are described in Section 3.2 as parts of the *Service Self-provisioning*.

In Section 3.3 and Section 3.4, the *Resource Manager* and the *Network Controller* functionality are described. In Section 3.5 and Section 3.6, some underlying technological components, the *Databases* and the *Interface Specifications* are described.

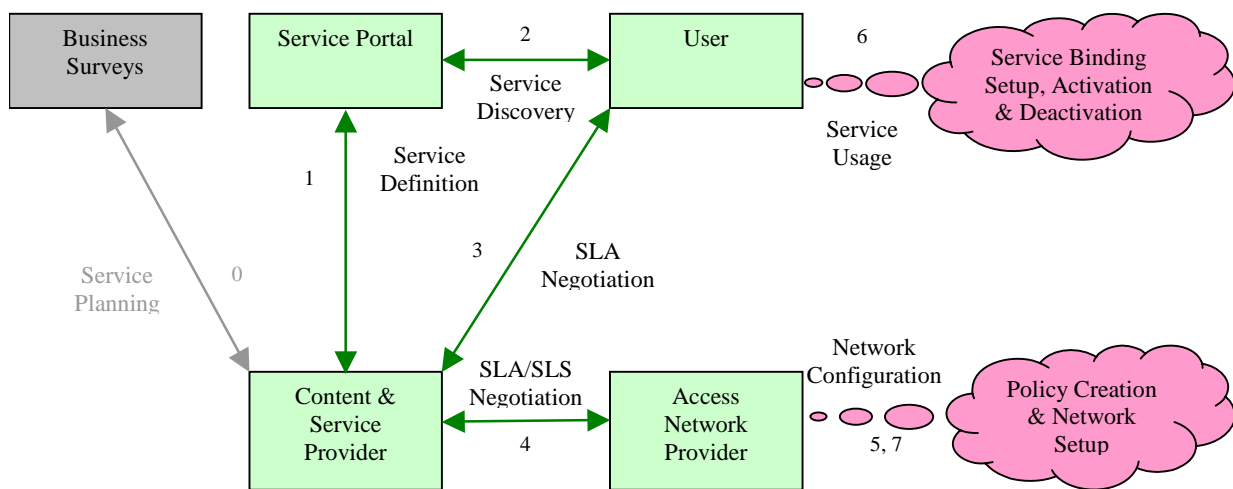


Figure 3-2 Whole Life-cycle of a Service

### 3.1 Service Definition

This section describes by who and how a service and its parameters should be defined. Three different levels of the definition are given and compared to each other. These levels are: normal user on SLA level, advanced user on SLA level and the SLS level. Normal user is not familiar with the technical details of networking, typically the subscriber. Advanced user has basic, general knowledge on service and network description parameters, typically an operator of the network provider or an expert of telecommunication networking. Finally parameters on the SLS level are mapped from the SLA to a particular number (or indicator set, etc.). This number is chosen from a set of parameter intervals in such a way that the SLS parameters should be stricter than the SLA, so if the network satisfies the SLS, then the SLA is automatically fulfilled. The parameters are general, so in particular cases some of them can be disregarded, on the other hand, the list may not cover all needs of a provider, so additional ones can supplement the list.

Based on the SLS parameters, the appropriate policies can be determined in the network concerning the given service. The most important policies are:

- The priority class which the service belongs to.
- The amount of capacity to be reserved for the service.

- The shaping and policy rules at both ends of the network (in the uplink direction at the Access Node, in the downlink direction at the Access Edge Node):
  - Admitted traffic volume.
  - Shaping and fairness queuing method.

The service parameters can be differentiated based on their permanence and their relation to the service lifecycle. This latter means that a parameter is “public” already in the *service definition phase* or just in the *service subscription phase*. Of course, the parameters belonging to the service definition phase are *constant, unvariable parameters* by the users like the name of the service. The parameters belonging to the subscription phase can be *inconstant, modifiable and selectable* (both mandatory and optional) like the speed of an Internet connection. This kind of differentiation of the service parameters is shown in Table 3-1.

Service parameter groups	Service parameters	Short overview of the parameters
Constant <i>service definition</i> related	<b>Name</b>	Name of the service, the name of the provider and a short description of the service.
	<b>Type</b>	Description of the service type with highlighting the main groups of the service, which it belongs to. (This helps to sort the services in service portal.)
	<b>Charging</b>	Specify the type of charging (e.g. flat, time-based).
	<b>Duration</b>	The lifetime of the service (e.g. one occasion, 1 month long, indefinite).
Constant <i>service subscription</i> related	<b>Addressing &amp; numbering</b>	Specify the type of addressing/numbering for the service.
	<b>Security</b>	Define the level of authentication before using the service and the way of monitoring the activity of the user. Furthermore, define the way of coding/decoding the data for secure transmission.
	<b>Control</b>	Define the level of self-provisioning meaning the ability for end user to add himself some options relevant to an already used service. Furthermore, the ability of remote control (e.g. in case of burglar alarm system).
	<b>Default connection</b>	Default edge node and default service agent through the service can be reached. Also describes whether the service is internally (e.g. local VoIP call) or externally provided (e.g. online casino abroad).
Modifiable <i>service subscription</i> related	<b>Traffic characteristics</b>	Description of the main characteristics of the traffic (e.g. real-time asymmetrical bidirectional point-to-point service).
	<b>Throughput characteristics</b>	Description of the behaviour and distribution of the traffic (e.g. constant or variable bitrate is needed by the service and how much capacity).
	<b>Quality of Service</b>	The observed QoS-requirements depend strongly on human perception of the application quality, so in many cases QoS should be regarded as more or less fixed to a standard parameter (e.g. the delay has to be below 200 ms in order to provide a pleasant telephone conversation).

Table 3-1 Overview of the main parameters of a service

In Table 3-2 parameters that are constant and cannot be modified by the user and apriori given by the service provider are listed. Here the most important constant parameters are listed from a *normal user's* point of view, which are related to the service definition phase. Those parameters, which are more technology specific and related to subscription, are listed in Section 4.2.5.

Service parameters	Sub-parameters	SLA		SLS
		Normal user	Advanced user	
Name	Name of the service	Text	Text	Text
	Name of the provider	Text	Text	Text
	Description	Text	Text	Not visible
Type	Residential / business	Text	Text	Not visible
Charging	Charging criteria with the applied costs	Time/Time of day (e.g. per minute)	Time/Time of day (e.g. per minute)	Time/Time of day (e.g. per minute)
		Volume/Transfer (e.g. per MByte)	Volume/Transfer (e.g. per MByte)	Volume/Transfer (e.g. per MByte)
		Distance (e.g. per km)	Distance (e.g. per km)	Distance (e.g. per km)
		Unit (e.g. per transaction)	Unit (e.g. per transaction)	Unit (e.g. per transaction)
		All-in price	All-in price	All-in price
Duration	Lifetime of the service	Length of the lifetime (1 day, 1 month, indefinite) or the number of occasions	Length of the lifetime (1 day, 1 month, indefinite) or the number of occasions	Length of the lifetime (1 day, 1 month, indefinite) or the number of occasions

Table 3-2 Constant service parameters related to the service definition

In spite of that the above parameters are constant, their actual value can be varied on the actual setting of the modifiable parameters. For example, if the user changes the speed of his Internet connection, then the charge of it should be automatically updated.

### 3.2 Service Self-provisioning

Since there are many competing service providers with a multitude of offered services, the service provisioning must be made quick and easy. The access network must aim to provide methods for self-service provisioning, such that users can freely select and get immediate access to any available services.

The trading entities of the access network are service bindings. The most obvious way of creating these service bindings is for the users to go to a service selection web-page and request a service binding with a particular set of parameters to a service, and then immediately able to start using it (unless the service provider needs to perform some measures before the service can be used).

This section defines and outlines how a service binding can be established, used, and terminated from a user device. It assumes that the user wants to get connected to a service provider's IP network. Our aim is to introduce the self-provisioning from the user's and from the access network's point of view. In the latter case, the layer 2 and 3 messages should be in the focus (from the user device to the access system and between the Access Node and the Access/Service Edge node).

The lifecycle of a service from the *self-provisioning* point of view can be separated into two parts. The first part is the *establishment* (including service discovery, SLA & SLS negotiation and input for the policy creation), typically done once, and the second part is the repetition of *activation, usage and deactivation* of the service (see Sections 3.2.1.1-3.2.1.3 for user- and Sections 3.2.2.1-3.2.2.3 for network perspectives). Furthermore two special aspects are introduced: the *use of a multicast service binding* and the *application of multiple service bindings by one host* (see Sections 3.2.1.4-3.2.1.5 for user- and Sections 3.2.2.4-3.2.2.5 for network perspectives and examples).

### 3.2.1 Self-provisioning from the user's point of view

#### 3.2.1.1 Establishment of a service binding

First most important steps to establish a service binding is presented from the user's point of view:

1. The user connects his/her "web capable" device (PC, palmtop, etc.), which gets a temporary IP address for the user.
2. When starting a web-browser, the user is directed to a service selection web page hosted by the service administration function.
3. The user selects the required service (concerned as a service discovery) and sets the service descriptive parameters (like Internet connection speed), hereby forming an SLA negotiation with the Content/Service Provider (practically checking the prospective costs according to the parameter/requirement setting) and also creating a service binding (after accepting the conditions of the service).
4. The "network" assigns an IP address to the user for the working session and sets his/her default gateway, router and DHCP server parameters also.

If the user wants to use another device in the above-selected service, then he just attaches to the network and the device should automatically goes through the above step 4.

For technical details please see Section 3.2.2.1.

Note that in case of IPv6 network scenario, the IP address of a user (or the IP address of his/her devices) can be given in advance, so the corresponding steps of the establishment can be skipped.

#### 3.2.1.2 Activation and use of service binding

In order to activate the service, the user device should resolve the IP address of the default gateway, then it is in a position to send and receive traffic. This action is practically hidden from the user.

If a message originated from a user's domain for which no service binding was established, the message triggers a service provisioning action, which is the establishment of a service binding.

For technical details please see Section 3.2.2.2.

#### 3.2.1.3 Deactivation of a service binding

If the user wants to deactivate his/her service binding prior to its expiration date, the user has to indicate it at the web portal, hereby his/her IP address will be made available to other users and the service binding associated to the user's domain will be deactivated.

In another case, the above procedure can be forced by the DHCP server if it does not hear from the user device by the end of the lease on the IP address and it marks the lease as non-renewed.

For technical details please see Section 3.2.2.3 .

#### 3.2.1.4 Use of a multicast service binding

From the user's point of view, the use of a multicast service binding is only a question of choice, namely if the bound service allows to send multicast messages (e.g. in a VPN service), it is done automatically after joining or leaving the multicast group.

For technical details please see Section 3.2.2.4.

#### 3.2.1.5 Multiple service bindings by one host

A user environment contains a number of devices using services accessed over a number of service bindings. Some of these devices are naturally (e.g. and "IP telephone") single service devices, needing only support for one service binding at a time, while others, such as PCs are, or can be expected to evolve to, multi-service devices where the user can e.g. both be connected to the internet and watch TV programs, possibly even simultaneously. For such devices communication using several service bindings must be supported simultaneously.

Let us assume that the user activates multiple services from different service providers by employing multiple service bindings to the same user device. In this example, the user has subscribed to Internet access and two video services. The goal of the OAM solution is that the user should not notice any difference compared to the activation of a single service. So the user first creates a service binding to the Internet Service Provider network after provisioning it through the service selection webpage. Next, the user initiates a VoD service supported by an Application Service Provider through the service selection webpage. Then the user starts a second application service, now another VoD service supported by another Application Service Provider, using the service selection webpage.

For technical details please see Section 3.2.2.5.

### 3.2.2 Self-provisioning from the network's point of view

#### 3.2.2.1 Establishment of a service binding

The most important steps to establish a service binding from the network's point of view are:

1. Find a DHCP server:
  - a. The device, when connected and powered up, sends out a broadcast message (DHCP DISCOVER) in order to find a DHCP server in the network.
  - b. The broadcast message is caught by the Access Node and handed over to its Broadcast Handler, which sends the DHCP DISCOVER message, together with the information of which Access Node and user port the request came from, to the Broadcast Handler at the Access Edge Node.
  - c. Then the DHCP DISCOVER message is passed to the binding control logic at the Service Edge Node, where it is determined that there is currently no service binding and no associated service agent (a service administration and handler functionality in the Service Edge Node) for the user's domain at this Access Node port. Therefore a default service agent called service administration service agent is appointed to take care of the request and to allocate a temporary locally administered (virtual) MAC address (LocAd-MAC address) for this user.

- d. The DHCP DISCOVER message is then handed over to the DHCP server associated with service administration service agent, with a request to respond with a DHCP OFFER.
2. Get a temporary IP address:
  - a. The DHCP server sends a DHCP OFFER unicast message to the device.
  - b. The DHCP server and the user device walk through the DHCP assignment sequence, resulting in that the device is configured to have a temporary private IP address, and with the IP address of the service agent as default router address.
3. Select a service:
  - a. *Service discovery*: The user is now directed to a service selection web page hosted by the service administration function.
  - b. *SLA negotiation*: There the wanted service provider and the bandwidth and other attributes of the service binding are selected.
  - c. *SLS negotiation*: The service selection is sent to the binding control function, which allocates a service binding to the service agent of the selected service. Based on the above accepted SLA, the SLS can be created for the given service and the user port and user's domain are associated to that LocAd-MAC address.
  - d. *Input for Policy creation*: all important parameters are sent to the policy management that determines where (in which policy enforcement points) and which policies (based on the policy repository) should be applied in the network in order to satisfy the SLS during the usage of the service.
  - e. The DHCP server associated with the service administration service agent now sends a unicast DHCP FORCERENEW message to the user device to invalidate the temporary IP address.
4. Get an IP address for the working session:
  - a. The user device now broadcasts a new DHCP DISCOVER message using a broadcast message.
  - b. The Access Node intercepts the broadcast message and hands it over to the binding control at the Service Edge Node through the broadcast handler at the Access Edge Node.
  - c. Then the binding control recognises the user's domain at this Access Node port to be associated with the service binding to the selected service agent.
  - d. The DHCP server associated with the selected service agent is instructed to respond with a DHCP OFFER.
  - e. The DHCP server sends DHCP OFFER to the user device and configures it with IP address belonging to the selected service provider network and sets the selected service agent as the default router.

After this the user device has access to the selected network and service. Additional devices attached to the same LAN or VLAN will need only to go through the point 4, since they reside in a user's domain which already has a service binding to the service provider's domain.

Note that in case of IPv6 network scenario, the IP address of a user (or the IP address of his/her devices) can be given in advance, so the corresponding steps of the establishment can be skipped.

### 3.2.2.2 Activation and use of service binding

In this case a service binding has already been established to a service provider's domain, that the user device has already been assigned an IP address belonging to the service provider's domain, and that the default gateway of the device has been set to the IP address of the appropriate service agent.

1. Before it is able to send any type of frames or IP datagrams, the user device needs to perform an ARP request to resolve the IP address of the default gateway to its corresponding LocAd-MAC address.
  - a. The user device broadcasts an ARPREQUEST message on its local physical subnet using the Ethernet broadcast address.
  - b. The broadcast frame is caught by the Access Node and handed over to its broadcast handler with the information on which user port the frame came from. The broadcast handler then tunnels the ARPREQUEST message to the Access Edge Node with the addition of an outer Ethernet frame, together with the information on which Access Node and user port the request came from.
  - c. The ARPREQUEST message is handed over to the broadcast handler of the specified service agent at the Access Edge Node with the intent to respond with an ARPREPLY message.
  - d. The ARP handler associated to the specified service agent replies with an ARPREPLY message. This message is unicast through the Access Node to the address of the user device, which sent the original ARPREQUEST message. Since the request included the MAC address of the user device, there is no need to send another ARP message to find it.
2. Once the user device has resolved the IP address of its default gateway to its corresponding LocAd-MAC address, then it is in a position to send and receive traffic to and from the service provider's domain, through the Access Node and Service Edge Node, using the established service binding.
3. By that time the service binding is considered as established, the resource management has to reserve the needed resources meeting the QoS requirements laid by the user.
4. Sending upstream traffic:
  - a. For upstream traffic frames, the MAC destination is the LocAd-MAC address established for that user, and the MAC source is the user device's MAC address. The IP source address is the user device's IP address originally given by the DHCP server of the service agent associated to the service binding. The IP destination address is an IP address belonging to the service provider's domain.
  - b. In the Access node, upstream traffic from the user port is only allowed if the destination MAC address is a LocAd-MAC address established for that user.
  - c. In the Access Edge Node, upstream traffic is only allowed from a user's domain for which a service binding was previously established, and where the MAC source address and the IP source address of the upstream frame are valid for that service binding.
5. Getting downstream traffic:
  - a. For downstream traffic frames, the MAC source address is the LocAd-MAC address. The MAC destination address is the user device's MAC address. The IP destination address is the user device's IP address originally given by the DHCP server of the service agent associated to the service binding. The IP source address is an IP address belonging to the service provider's domain.

- b. In the Access Edge Node, downstream traffic to a user domain is only allowed if the destination IP address and the service provider's domain can be used as a unique key to get the corresponding LocAd-MAC address and the MAC address of the user device.
  - c. In the Access Node, downstream traffic to a user domain is only allowed if the MAC source address in the downstream frame is a LocAd-MAC address established for that user.
6. If the message originated from a user's domain for which no service binding was established, the message triggers a service provisioning action in the Service Edge Node.

### 3.2.2.3 Deactivation of a service binding

In this case a service binding has already been established to a service provider's domain, the user device has already been used the service and now wants to finish it.

1. If a user device finishes using an IP address prior to its expiration date, the user device sends a DHCPRELEASE message to the DHCP server associated to the service agent so that the IP address can be made available to other user devices.
2. In the Access Edge Node, after receiving the DHCPRELEASE message, the DHCP server interacts with its related service agent to deactivate the service binding associated to the user's domain.
3. If the DHCP server does not hear from the user device by the end of the lease on the IP address, it marks the lease as non-renewed, and makes it available for other user devices to use. It also interacts with its related service agent to deactivate the service binding associated to the user's domain.

### 3.2.2.4 Use of a multicast service binding

When the user device decides to join or leave a multicast group, it sends its IGMP messages to the access network. The Access Node, which recognizes these types of messages by their multicast address to which they are sent, will capture these messages and tunnels them to the Access Edge Node, to be handled by the service agent handling the multicast. The service agent will then analyze these messages and send back instructions to the access node to send relevant IGMP messages in order to join or leave the multicast group.

### 3.2.2.5 Multiple service bindings by one host

In our concept, each service binding has a unique MAC address used as the MAC address of a service agent. Depending on the type of service binding and the relations between the different service providers, the LocAd-MAC address will be associated with:

1. The MAC address of the default gateway.
2. The MAC address associated with a host address of an Application server.
3. The MAC address of another gateway address, occurring in the routing table of the attached host.

### 3.3 Resource Manager

The delivery of QoS for each connection in network involves building a view of the available resources and controlling the use of the network accordingly. The Resource Manager implements these functions. The N-RAF has been designed to be independent of the physical network. We propose a centralized network resource management mechanism for network resource control.

The centralized resource management has its pros and cons, namely:

<b>Pros</b>	<b>Cons</b>
Better network utilization (avoids partly reserved path)	Requires signalling for decision
Central view of network resources	Signalling introduces delay for service setup
no need to build states in the forwarders	Requires accurate view of network resources

The reason we propose a centralized management is that for billing, security and management purposes the service request procedure involves a centralized management which can be extended to support admission control. Thus, the Central Network Resource Manager (CNRM) which can be a responsibility of the Packager. The service setup (service binding) and admission control should be implemented to work together, avoiding unnecessary signalling and delays. Thus, when the service is requested, not only AAA should be performed but also a Call Admission Control (CAC) decision should be made. Scope of the central CAC is the complete access and aggregation network. This is in line with the centralised CAC described in [Ref 13].

The proposed resource management combines the CNRM concept and using resource pipes in the aggregation network. The pipe concept adds flexible management of the resources, and makes the admission control simple.

Admission control is necessary if we would like to gain benefits of statistical multiplexing while maintaining QoS guarantees. In case of an over-provisioned network the admission control of course is completely useless.

#### 3.3.1 Admission control mechanisms

The basic admission control is the case, when the network is dimensioned for all users requesting the services simultaneously. In this case the admission control is done when the user subscribes to a service: so if he has a subscription, he can use the service at any time. This basic approach results in a pre-engineered network, such as nowadays' telecommunications networks. This method is not suitable for provisioning high bandwidth on-demand services, but fits for the virtual leased line services.

The admission control for on-demand services requires the following steps:

1. The user requests the service. In our case, this means that the user clicks on a service on the list of services or requests a service using application signalling etc.
2. The user's right verification that the user really has permit to use the requested service. If not, the request should be rejected. If he has permit the Service Administration Function (SAF) determines the QoS requirements, which is part of the SLA negotiation.

3. The Network Resource administration Function (N-RAF) verifies if there are enough resources in the network to admit the new flow(s). If not, the request should be rejected.

It is important to be able to signal to the user the reason why the admission failed:

- authentication problem
- permission problem (user requesting service he didn't pay for)
- temporal lack of resources
- service is temporarily unavailable

4. The resources must be allocated for the new flow in the resource management system. Parameters for policing must be properly set in the edges.

5. The service binding should be done for the new service.

6. The service can now be started. In case of policy push option the user receives a start message.

### 3.3.1.1 Simplified connection admission control (CAC)

The QoS part of DA2.4 [Ref 13] states that CAC accuracy may vary from very stringent CAC for low rate access links down to “no CAC” on over-provisioned backbone links. For the MUSE QoS class preferred implementation, two simplified CAC algorithms have been derived: one for low rate interfaces (<10Mb/s) and one for high rate interfaces (>100Mb/s), called slow and fast CAC, respectively.

Basic idea of the MUSE QoS preferred implementation is to have very few parameters to specify when a flow (a service binding) is configured. As described in [Ref 13] MUSE defines four QoS classes only. As a further simplification each class has one dynamic parameter only, the guaranteed rate. All the other parameters such as maximum packet/ burst size, delay, jitter as well as DSCP and P-bit values are fixed for each class (see Table 3-3).

QoS class	DiffServ code point	Max. burst size	Max. Node jitter	Mean Node jitter	Max. Packet loss rate per node	P-bits
Low Latency	101 010	200 byte	1 ms	0.04 ms	$10^{-10}$	110
Real Time	101 110	1500 byte	30 ms	1 ms	$10^{-10}$	101
Elastic / Flexible	100 010	9000 byte	900 ms	36 ms	$10^{-10}$	011
	100 110					010
Best Effort	Any other					000

Table 3-3 QoS class parameters

Furthermore it is proposed to specify the guaranteed rate in multiples of 100kb/s, which again simplifies the specification of the QoS of a flow (service binding). Ultimately 12 bit would be sufficient to identify the service binding in the aggregation network, 10 bit for the multiple of 100kb/s and 2 bit for the class.

Some of the S-VLAN ID-s must be reserved for multicast flows, e.g. IPTV channels. To efficiently utilize the network resources in the aggregation network, VLANs should be used to implement multicast service flows.

Both slow and fast CAC are described in [Ref 14]. The slow CAC has been derived by a deterministic approach taking the number of flows into account.

1. CAC for low speed link (<10Mb/s)

It is based on a deterministic approach. Its application is 1<sup>st</sup> mile links. For derivation and detailed description see DTF3.2. The acceptance algorithm for a new flow (or service connection) is shown in the figure below. One can see that higher level flows affect lower level flows, but not vice-versa. For example a new Low Latency (LL) flow could be rejected although bandwidth is available and the number of flows in the LL class is not exceeded.

Each flow is specified by its guaranteed rate – which may be zero in case of Elastic/ Flexible (EF) or Best Effort (BE) class. The difference is that for EF the number of flows is checked, whereas for BE class it is not. The decision “Bandwidth exceeded?” in the figure means that the sum of guaranteed rates plus the guaranteed rate of the new flow is still below the maximum rate of the link. The sum of guaranteed rates, the three numbers of maximum allowed flows and the three numbers of currently allowed flows must be maintained by the bandwidth management.

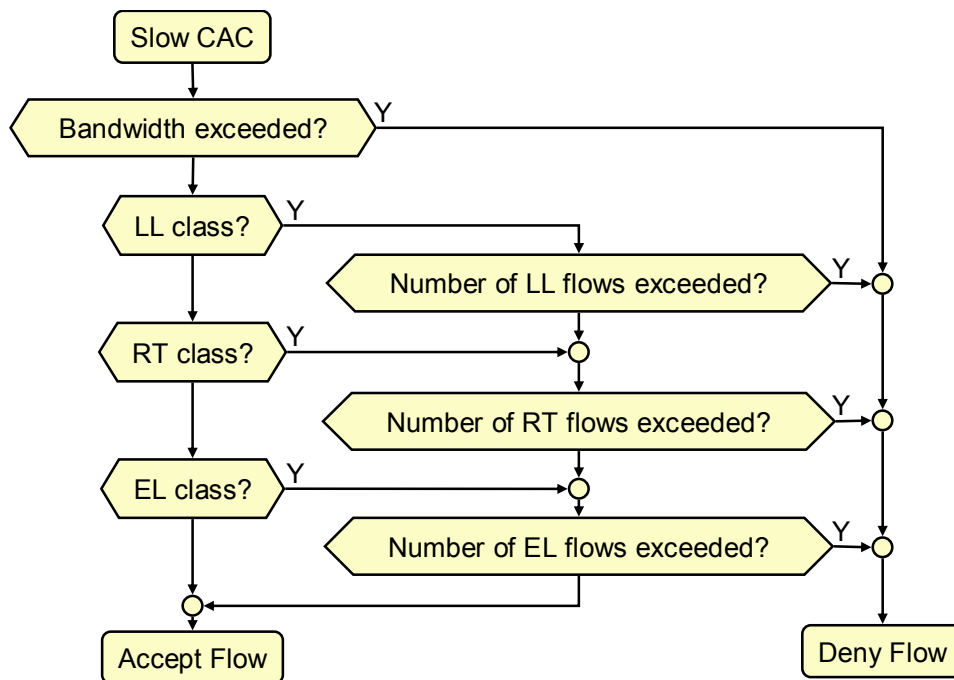


Figure 3-3 Slow link CAC

2. CAC for high speed links (>100Mb/s)

This is based on a statistical approach. It checks only the sum of guaranteed rates as shown in Figure 3-4.

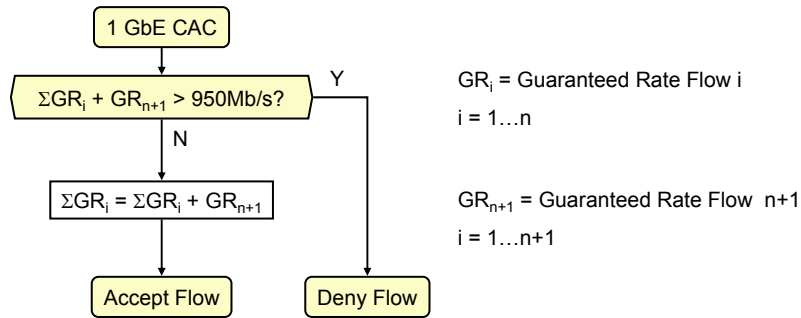


Figure 3-4 Fast Link CAC

Other than in low speed CAC maximum bandwidth is below the link rate as shown in Table 3-4 for some selected link rates. In this algorithm EF with zero guaranteed rate and BE are treated identically.

Line rate	Maximum load %	Maximum Load
	of Guaranteed traffic	
100 Mb/s	57%	57 Mb/s
150 Mb/s	72%	108 Mb/s
600 Mb/s	93%	558 Mb/s
1 Gb/s	95%	950 Mb/s

Table 3-4 Maximum Load for Fast Link

To simplify the high speed CAC a statistical approach has been made based on the M/D/1 (The first part represents the input process, the second the service distribution and the third the number of servers) [Ref 5] queuing model. For each QoS class the maximum load has been calculated using the respective parameters maximum packet length, maximum jitter etc. The obtained maximum load values are listed in the table for some typical backhaul links:

Line rate	Maximum load Low Latency class - Node Jitter 1ms - Packet size 200byte	Maximum load Real Time class - Node Jitter 30ms - Packet size 1500byte	Maximum load Elastic class - Node Jitter 900ms - Packet size 9000byte
100 Mb/s	57%	92%	98%
150 Mb/s	72%	95%	98%
600 Mb/s	93%	98%	98%
1 Gb/s	95%	98%	98%

Table 3-5 maximum load per QoS class

---

For final simplification the maximum load of the most critical class is taken for all QoS classes – in this case the values of the Low Latency class.

### 3. CAC for very high speed links (>1Gb/s)

It can be called “null-CAC”. At flow setup it is only checked if overload state is reached or not. No bandwidth sum must be maintained. However, there are two problems to be solved, how to determine overload state and how to convey the overload state from management plane to control plane. The first item is related to network topology and routing. For example one of more links could be in overload state and a new flow is requested, then the question is if the new flow passes the overloaded links. As a simplification an overall network load could be derived by the management plane from individual link loads and conveyed to the control plane. This item is for further study.

To determine the overload state of a link the guaranteed packet flow is measured for example over a period of 1s. Non-guaranteed packets are ignored. If for 10 consecutive seconds the load exceeds the maximum allowed value the link is declared overloaded. Conversely after 10 non-overloaded seconds the link is declared available again. In case of overload further flow setup requests over that link are blocked.

Alternatively a method has been proposed where not only the overload state is signalled, but the current load. Depending on the priority of the respective service flow setup request could be rejected if the load exceeds an associated threshold. For example a bandwidth consuming video service could be rejected when the load exceeds 70% of the allowed value, whereas beyond 95% only emergency phone calls could be accepted.

A new flow can be accepted regardless of its QoS class, as long as the maximum load for the respective link rate is not exceeded. Link usage on 100Mb/s links seems to be poor. However, this rate limit only applies to the guaranteed portion of the traffic. The remaining bandwidth will be used by low priority Elastic/ Flexible flows (statistical multiplexing) and for Best Effort traffic. With the trend towards 1Gb/s links the penalty is negligibly small.

The simplified CAC approach can be applied also to virtual pipes with rates between the discrete values of Table 3-5. In these cases the maximum load can be obtained by interpolation.

Note that the consequences of the simplified CAC proposal based on the MUSE QoS class preferred implementation are far reaching:

- Only one pipe must be reserved for all QoS classes.
- Only one parameter must be kept for CAC: the sum of guaranteed rates.
- QoS class is linked to p-bits and decoupled from route information (MAC, VLAN).
- OAM connectivity fault management packets can be sent for each QoS class with the respective setting of p-bits.

There are several options identifying resources in the aggregation network. Per-node resource reservation is not suitable, since it will highly increase the complexity of the nodes in the aggregation network. In the SPC central resource management method is proposed. The following scenarios are considered: Traffic Engineered Pre-provisioned pipes, Traffic Engineered un-provisioned pipes, Central resource management with spanning tree protocol query and on-demand routed. These will be elaborated in the next four sections:

### 3.3.1.2 Traffic Engineered Preprovisioned pipes

For each Access-Edge pair and class we define a provisioned pipe. Each VLAN pipe {access node – edge node} should be identified by a VLAN ID (S-VLAN ID). Multiple VLAN IDs could be used for a single pair {access node – edge node} in order to differentiate in terms of service or QoS a more scalable alternative is to define a unique VLAN ID per pair {access node – edge node}. The p-bit would indicate the received QoS class according to the values defined in DA2.4 [Ref 13].

This should be done in advance, defining a table that holds the maximum available bandwidth that can be allocated for each S-VLAN, that identifies a pipe between an AN and an EN for a specific service. The provisioning of the pipes is done once by the network administrator, based on network topology and link speeds.

The admission control checks if there remains enough bandwidth in the pipe, and if admits, subtracts the bandwidth from the available bandwidth of the pipe. The AC can be done at the AN by the CNRM entity.

The advantages of this method are the simplicity and that the Spanning Tree protocol defines a tree topology and therefore it is relatively simple to fill in the table.

The disadvantage is the lower network utilization since if one access-edge-class triplet does not use all of its dedicated bandwidth the others are not permitted to take it over. More precisely some service bindings could be not admitted in one pipe although other pipes sharing the link(s) have spare capacity. However, due to statistical multiplexing low priority packets of Elastic/ Flexible class and Best Effort class packets can still use the bandwidth up to 100%.

The applicability with demo concept is quite straightforward, since VLANs are used mostly in the same sense.

### 3.3.1.3 Traffic Engineered un-provisioned pipes

In this case we set up a path for all Access-Edge Class pairs and store in a table the links used by this path. This path can be referred as an un-provisioned pipe, since we do not assign bandwidth for it. The bandwidth of pipe is not shared among services or service classes. When making admission control, for each link of the path we check if there is enough bandwidth in the given QoS class. We can only admit the new request if on all links of the path there is enough bandwidth available. After admission, we subtract the requested bandwidth from all links on the path. If the MUSE QoS class preferred implementation is used, bandwidth check must not be class specific (see fast link CAC Figure 3-4). Figure 3-4 shows an example for traffic engineered un-provisioned pipes. The un-provisioned pipes are set up on a hop-by-hop basis, and their path can be traffic engineered. The coloured lines show the path of a given pipe. The bandwidth for the pipe is not reserved, but for each demand it is accounted on each link, and registered at the CNRM.

The advantage of the method lies in that it can use the available bandwidth in a better way. The disadvantage is that we must define the path for each Access-Edge pair, and it requires more complicated admission logic, since all links over the path must be checked from the database. Other disadvantage is that in case of link failure the QoS guarantees may be lost.

Basic examples have been described in [Ref 29,Ref 30]. It could be an exact continuation of our work done regarding QoS and traffic engineering.

### 3.3.1.4 Central resource management with spanning tree protocol query

The admission control is done on per-link basis, but the path information is collected from the running spanning tree protocol. The collecting can be done periodically by SNMP or by means of OAM (configuration management).

The advantage over the previous case is the simpler management and in case of failure recovering by the Spanning Tree. However, in order to deploy this method in the network the interaction with the spanning-tree protocol is required. The optimal way of doing this is still under investigation.

### 3.3.1.5 On demand routed

This method can be used in case when MSTP based traffic engineering permits multiple paths between Access-Edge pairs. Besides a default path between Access-Edge pairs one or more alternate paths are defined using multiple spanning tree instances. This requires that for each path a different VLAN-ID is assigned and the VLAN belongs to a different MST instance [Ref 31]. The CNRM keeps track of all resources, and when the default path is congested, an alternate path is selected to carry the traffic. In this case the admission control provides VLAN-ID to be used. This VLAN-ID selects the desired path for the admitted flow. This method could provide the best results for dynamically changing network load.

An example is shown in Figure 3-5 and Figure 3-6. It shows a network which has two different VLAN topologies, green and blue. In Figure 3-5 one backhaul link has reached its maximum capacity. Hence for all further connection setup requests the central admission control selects VLAN blue. In this topology the overloaded link is not used.

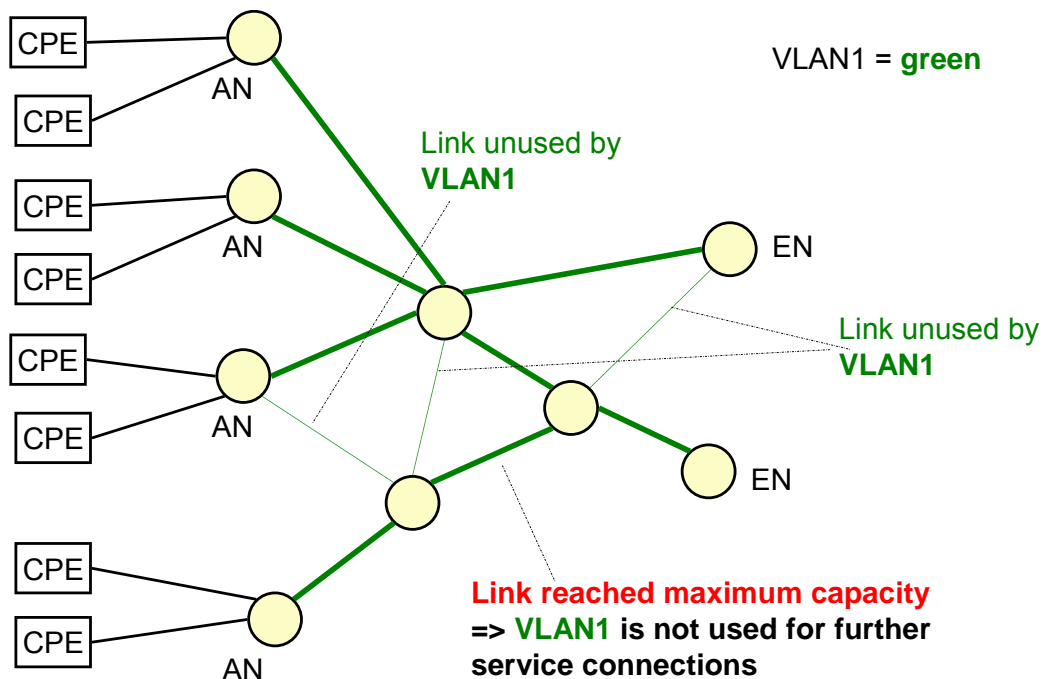


Figure 3-5: Example network with VLAN1 (green)

After a certain time capacity might be free again on the overloaded link of VLAN green, whereas a link of VLAN blue could be blocked. Then the admission control would switch back to VLAN green. Further VLANs could be configured, so that blocking becomes more and more unlikely.

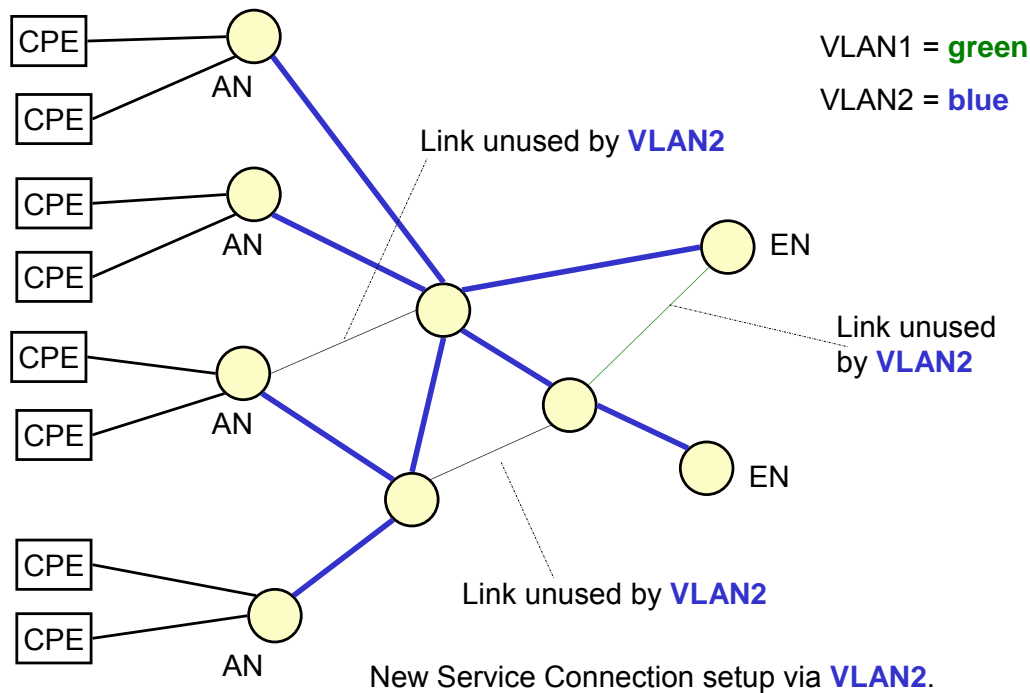


Figure 3-6: Example network with VLAN2 (blue)

The traffic engineered un-provisioned pipes model has the advantage of better network utilization at the cost of algorithm complexity at the CNRM: it must keep track the resource usage on each link in the domain. However, routing is not performed for each request, thus the scalability of the algorithm is not questionable.

The required link state database can be maintained by querying the spanning tree protocol, however this is a further extension of the method.

### 3.4 Network Controller

The Network Controller (NC) controls the network as a whole. It allows network devices from different vendors and with different capabilities to be configured properly. So, the network controller provides all the needed technology-dependent functionality. It's main role is to translate network independent service requirements from the Resource management level into network dependent requirements or policies (NDP). It configures and manages the underlying network devices by the NDP to provide the contracted services for end users.

The NC receives bandwidth reservation request from the N-RAF in the resource reservation phase and put changes to the Resource Database (RDB).

---

The NC communicates with the Device Controllers (DC) through the Policy Enforcement Point PEP, which controls network elements like aggregation switches and access nodes. The DC has to set some leaky-bucket parameters (like rate limiting and traffic shaping) for devices on the selected path for the flow.

## 3.5 Databases

### 3.5.1 Global Service Registry

Global service registry contains all services from different service providers that are available to end-users. The registry can be managed or even owned by an entity playing packager's role. The Universal Description Discovery and Integration (UDDI) registry technology is recommended for this registry.

#### 3.5.1.1 UDDI Registry Technology

The UDDI protocol is a central element of the group of related standards that comprise the Web services stack. The UDDI specification defines a standard method for publishing and discovering the network-based software components of a service-oriented architecture (SOA). Its development is led by the OASIS consortium of enterprise software vendors and customers.

A UDDI registry's functional purpose is the representation of data and metadata about Web services. A registry, either for use on a public network or within an organization's internal infrastructure, offers a standards-based mechanism to classify, catalog, and manage Web services, so that they can be discovered and consumed by other applications.

UDDI is based upon several other established industry standards, including HTTP, XML, XML Schema (XSD), SOAP and WSDL. The core information models used by UDDI are defined by XML Schema.

#### 3.5.1.2 Global Service Registry: function and data model

Global service registry acts as a repository of registered services and provides the end-users with the necessary lookup facilities. To support the equal access business model, different services with diverse QoS characteristics need to be represented and supported in the registry. To make that happen, new data models, e.g., QoS characteristics, need to be defined in the registry. One way of incorporating QoS into the UDDI model is to include QoS as attributes when services are published by service providers to the registry and take it as constraints when services are searched by end-users.

### 3.5.2 Service Depository

Service depository is used to store service related informations that belong to individual service providers before they are published to the global service registry, and also after their publication to the global service registry as a backup.

---

Services stored in the service depository are not publicly available to end-users. From that perspective, data model for service depository can be anything as long as the service provider that provides the service feels comfortable with it. However, since services in the service depository will eventually be published in the global service registry, it is recommended that the service depository also use the UDDI registry technology so that integrity can be easily maintained between the service depository and global service registry. When using the UDDI data model, all services in the service depository will have the same service provider name.

### 3.5.3 SLA/SLS Database

SLA is a contract between the end-user and the service provider that defines a set of characteristics of the service delivery. SLA databases are usually managed by service providers and the data structure of SLAs stored in the database should be text based and human readable. The following information needs to be included in an SLA:

- Technical aspects, such as QoS attributes and availability of the services that the user asks for. The QoS attributes can well be a sub-set of the QoS parameters defined in the global service registry, depending on the relationship between the end-user and the service provider.
- None-technical aspects, such as pricing, penalties or verifying methods.

The end-users' expectation of the service is also specified in the SLA. To meet these expectations, the service provider plans the provisioning of resources for the requested service taking into account the agreement.

While the SLA represents a high level description of a required service, the Service Level Specification (SLS) is a more formal technical document containing a list of technical parameters used for the provisioning and reservation of the necessary network resources. The adoption of a user-side document as well as of a corresponding network-side translation is mainly due to the need of holding a separation between service subscription and the actual service implementation. Furthermore, while SLA subscription can be considered as a quite static process, SLSs are involved in a more dynamic process because of variable network conditions. For example, for two different subscriptions related to the same service, different SLS instances can be derived according to the current network resource status. Data model of the SLS database should also be text based and human readable, and it is one step closer to network policies than SLA. For instance, QoS for a service in SLA may simply be described as "gold" or "silver", while in SLS, this "gold" or "silver" will be mapped to a number of parameter values such as bandwidth, maximum delay, jitter, etc. A more detailed description of SLA/SLS management, including mappings between them, can be found in 4.2.

### 3.5.4 Policy Information Base

The policy information base is defined and specified in [Ref 3]. It is closely connected to network controller in the system and is mapped from SLSs. The data model should follow the specification of [Ref 3], while instances may vary depending on the different access network the system is built for. The policies in a PIB are stored using a tree structure.

### 3.5.5 User Directory

Users that logon to the self-provisioning portal do so to manage their services and identities. This may be accomplished through a single sign-on architecture or more traditional means. This specification does not adhere or propagate a prefer method. A common factor, however, is that user information needs to be maintained in a user directory.

The user directory is a central directory containing user-related information and is located in the SPANF. The type of user information stored in the directory is implementation specific. But the type of information that may be stored in it is among others: identity, profile, private policies, service subscriptions, billing, location, and security. If single sign-on is supported then this information could be utilized by the SPANF to allow users to centrally manage their identities and services registered with different service and content providers. Otherwise it may be used to perform basic service and identity management.

Information in the user directory can be accessed through several different interfaces. A user accessing a service or logging on to the self-provisioning portal may request information to be displayed through a web-based interface. Whereas SAFs may request user related data or authentication through a Web service or Diameter interface.

The structure and format of the user directory is implementation specific, but it is most likely that it will be based on LDAP. More important is that the interfaces between other SPANFs and SAFs are well defined to allow for mobility and interoperability.

### 3.6 Interface Specifications

This section will identify and describe the interfaces defined in the proposed OAM architecture based on Figure 3-7.

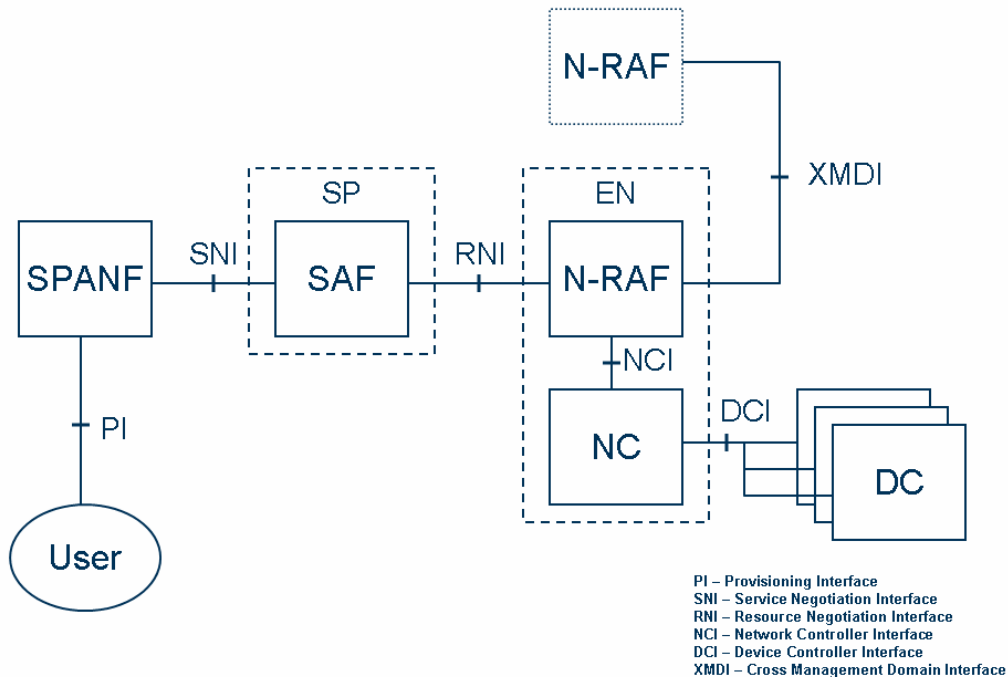


Figure 3-7 Interfaces

### 3.6.1 Provisioning Interface

The initial interface between a customer and a service provider is proxied by the self-provisioning portal hosted by SPANF. The self-provisioning portal provides a mechanism by which service providers can advertise their services. It provides a central forum where customers can go to browse and order services, manage services that have already been purchased, manage their online identities, and mechanisms to negotiate and manage SLA's with service providers. The provisioning interface provides a reference point between end-users and SPANF that offers the facilities to accomplish these tasks.

### 3.6.2 Service Negotiation Interface

The service negotiation interface is used to proxy the interaction between end-users and service providers and for service providers to publish services on the self-provisioning portal.

SLA negotiation information and user information are transmitted between the SPANF and SAF when a customer orders a service. Typical information sent during this phase includes parameters such as the service ordered, the availability of the service, customer preferences, user authentication and accounting.

Service providers shall be able to notify SPANFs of the availability of their services. This is to say if a service provider no longer has the capacity to provision services to additional customers or if specific resources are no longer made available. If a service is no longer available the SPANF should notify this information on the self-provisioning portal and to any service negotiation that is being processed.

SNI shall provide means by which service providers can publish and manage services at the SPANF. New services that are published may need to be approved by the SPANF before they are made public. This decision shall be sent over SNI to the SAF.

### 3.6.3 Resource Negotiation Interface

The resource negotiation interface (RNI) is used to exchange QoS resource reservation information between the SAF and N-RAF. RNI provides a method by which a SAF can send service-related QoS characteristics in order to request network resources, and means for an N-RAF to indicate admission control decisions.

RNI contains mechanisms to support resource management tasks between SAFs and N-RAFs, such as resource reservation and admission control messages, which should result in a mutual SLS agreement. This interface will transmit resource-constraint-based admission control messages between SAFs and N-RAFs, which will be forwarded through SNI to a SPANF.

It is important for the service providers to be aware of the status of SLS fulfilment because it is directly correlated to the realization of the SLA contract with end-users. RNI shall provide means for reporting network errors or states where an SLS is not fulfilled or when reserved resources have been revoked or are no longer available.

### 3.6.4 Network Controller Interface

When network resources have been negotiated and SLS's have been defined and the time comes to enforce a service then the network controller interface (NCI) is used to send service related policy information between the N-RAF and network controller. The network controller will translate this information into network specific policies that will later be deployed to devices on the network where they will be enforced.

NCI shall provide a manner by which policies can be managed on the network controller by the N-RAF. It shall be possible to push, pull, install, update, and remove policies through NCI.

It shall be possible for the N-RAF to obtain statistical data related to the operational status of the network controller. The interface shall allow the network controller to synchronize with an N-RAF in the scenario where configurations may have been compromised.

RNI must provide high reliability and information integrity in order to provide real time policy information in the network.

### 3.6.5 Device Controller Interface

Network controllers translate abstract network independent SLS policies sent by the SAF into network dependent policies. These policies are pushed to the network devices over the device controller interface (DCI) where they are enforced by the device controller. Additionally, the interface is utilized to transfer information about network devices, i.e. current status, and load, to the network controller that will update the policies running on the network devices.

Network dependent policies shall be sent through DCI from the device controller to network devices. DCI shall provide means by which the network controller can verify that policies are properly enforced by a device controller. This may either be accomplished by the queries sent from the network controller to device controllers or by device controllers contacting the network controller with policy enforcement reports. The latter may be most effective since it can be event driven. The most important information the network controller requires is the extent to which policies are enforced. Therefore, messages should be sent when device controllers realized that a policy has been breached.

DCI should be an open interface that allows the network controller to communicate with disparate vendor network equipment.

Mutual authentication mechanisms must exist between a network controller and device controller that prevent a third-party from sending false policies to network devices.

DCI should provide a reliable connection with high integrity in order for efficient end-to-end QoS delivery. If the interface is down, then the network controller can no longer push policies to network devices or receive network information that may lead to the N-RAF breaking the SLS with the SAF, and a breach in the SLA contract between the SAF and end-user.

### 3.6.6 Cross Management Domain Interface

Some scenarios require data traffic to traverse several network provider domains before reaching the end-user. This case requires the QoS can be negotiated in the intermediary domains in order to fulfil the E2E QoS requirements specified in the SLA. The cross management domain interface (XMDI) handles this negotiation process between N-RAFs in disparate domains as a consequence of the suggested Cadenus model. Network resource availability, pricing, and negotiation messages are exchanged through XMDI. It may also be used to setup trade agreements between two network providers (B2B). XMDI has similar qualities to RNI. What makes XMDI unique to RNI is that it contains mechanisms for bidirectional resource request messages. There are two possible methods to achieve cross management domain resource negotiation: hub and proxy.

In the hub approach, the SAF is able to identify all of the N-RAFs that need to be contacted in each domain that is traversed when delivering a service. The SAF then directly negotiates a SLS with each of the N-RAFs. The service can be delivered when the sum of the negotiated SLSs meets the minimum requirements to deliver the service as per the SLA. There must exist an N-RAF look-up functionality in order for this to be realized. A SAF must be able to localize each domain that the service will cross and which N-RAFs must be contacted for SLS negotiation. This approach is possible only if there is an entity that is able to identify the end-to-end route a service will need to travel and the N-RAF that is responsible for each traversed domain. A hub approach will require the SAF to have a business relationship with all of the network domains involved in delivering services to all of its customers. This solution may prove to be too complex on a large scale.

A second alternative is to use a proxy approach, as suggested by Cadenus. A SAF interacts with the N-RAF located in its local domain. The N-RAF is able to identify the route that a service must take to get to its location, and is therefore able to identify the neighbouring N-RAF. The SLS is split into two parts, one part for the local domain and a second part for the following domains. This approach scales better than the hub approach because it only requires the SAF to have a business relationship with the N-RAF in its local domain. The remainder of the negotiation process is handled by the N-RAF. Each N-RAF will only need to be aware of the N-RAF in each of the bordering network domains.

## 4 SERVICE DELIVERY

### 4.1 Resource Management

Resource management can be implemented in two ways:

- static
- dynamic

#### 4.1.1 Static management

The static management requires topology and routing information about the network. It also presumes that the routes in the network are also static. Based on the initial information all active connections are recorded in the database and the current resource status is computed. Upon termination, the terminated connections are removed from database, updating the bandwidth information. This way, the bandwidth broker has an up-to-date view of the network resources. This solution is simple, however at any change in network topology or routes will result in deterioration of quality of service.

---

In case of a link failure the STP protocol and the OAM both will detect the loss of connectivity. STP automatically looks for alternate paths, and if there are any available, it will restore the connectivity. The QoS requirements are not taken into account by STP, and also the management system is not aware of the new topology. However, the OAM will alert the administrator and human interaction is needed to correct the topology and routing database.

#### 4.1.2 Dynamic management

In this case the network topology and/or route information is acquired and periodically updated using SNMP or other means. The active connections are again recorded into a database and the current network load can be determined.

At any change in the network configuration resources can be recalculated and the resource manager can adapt to the new network status.

The following information is needed for dynamic management:

- connectivity information
- link capacity information
- spanning tree information

All of this information is available by means of SNMP, however it is possible to provide topology and capacity information in advance, using SNMP to gather spanning tree information only. A LoopBack qualification tool can be incorporated into the system for this purpose.

The Definitions of Managed Objects for Bridges (RFC 1493), Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions (RFC 2674) standards describe the MIBs required.

In case of link failure or topology change, STP automatically restores connectivity by searching alternate paths. OAM also detects the failure/topology change, and in this case triggers a topology update. Thus, the management system will update the topology by active polling (SNMP, etc). The OAM also informs the administrator about the change, but no human interaction is needed to update the topology database.

In both cases STP initiates the recovery, and OAM signals the failure/topology change to the network management. The network management is responsible to update the topology database in order to avoid further QoS degradation. The topology update is manual for the static case and automatic for the dynamic scenario. The failure signaled by the OAM is also an input for the Service Assurance framework.

#### 4.1.3 Bandwidth Brokering

The Bandwidth Broker is the part of the resource management system which makes the decision of admission or rejection based on the current network resource information and request parameters. The bandwidth broker may implement several admission control algorithms for optimal network utilization and can also provide a path on which the request can be admitted (if multiple paths available). Furthermore, in case when a request can be admitted only with QoS degradation, the negotiation of new QoS parameters is also the task of the BB.

Functionally the bandwidth broker is a software component with access to the resource management database. Its input is the new request (source, destination, traffic parameters) and its output is a decision (with or without a route).

#### 4.1.4 Admission Control

The admission control function is a signalling mechanism that communicates the client the decision. It can be either a stand-alone function or an integrated function with the service request. The advantage of the integrated function is the reduced signalling, thus lower service set-up delays.

In case of implementation it is desirable to use an integrated admission control. Three functionalities are required in this way:

- 1.) After the AAA, the AC functionality should extract the requested connection resource requirement information, and provide it to the bandwidth broker.
  - 2a.) In case of acceptance, the network resource management subsystem must be notified about the admitted flow
  - 2b.) In case of rejection, the client must be notified with the proper connection setup error message. This should be included in the service binding error message.
- More complicated scenario is when the bandwidth broker negotiates the QoS parameters.
- 3.) A third functionality is required when the service is terminated. The network resource management must be informed of the terminated connection.

#### 4.1.5 Support of Absolute and Relative QoS

The QoS support has two variants: absolute QoS and relative QoS. Absolute QoS specifies a set of parameters, such as bandwidth, packet delay, etc. Service with absolute QoS support will get the resources/treatment specified in the parameter set when the network can handle it, or will be rejected when the network does not have enough resource. Relative QoS, on the other hand, only specifies the relative importance of a specific kind of service by priorities. Service with higher priority will be treated more favourably than the ones with lower priority, however, there is no guaranteed resources to any QoS. In other words, the quality changes with the network status. During implementation admission control assure the resources for absolute guaranteed services, while relative services needs no admission control.

Both absolute and relative QoS are needed in the broadband access network. For services with strict resource requirement, such as IPTV, the absolute QoS can be applied, while for elastic services, such as VoIP and internet surfing, relative QoS can be used to differentiate them.

To support absolute and relative QoS at the same time, the p bits in the VLAN tag can be further categorized as following:

P bits field: xxx

Where 1xx indicates absolute QoS support, and 0xx indicates relative QoS support. This way, 4 absolute QoS classes can be defined with different definite values for bandwidth, packet delay, etc, and 4 relative QoS classes can be defined with different priorities.

Along with the QoS preferred implementation and considering QoS guidelines three CAC mechanisms have been derived, more detailed description of MUSE QoS class and CACs can be found in Section 3.3.1

The QoS can be negotiated between the user device and the service provider, or delegated by a third party device. After the QoS negotiation is done, the residential gateway will know how to fill in the p bits value for the upstream traffic of the service, and the access edge node knows how to fill the p bits of the downstream traffic.

In the access network, the connection between an end user device and a service provider is the service binding, which is maintained by a service agent. One service agent uniquely corresponds to a VLAN in the access network. The relationship between service agents and service providers are one-to-one or one-to-many, that is, the service provider can choose to set up only one service agent representing himself in the access network and let all traffic between the end users and the service providers flow in the same VLAN, or he can set up different service agents, thus different VLANs, for different services he provides. However, different service providers will never use the same service agents. Thanks to the guaranteed rate concept services with absolute QoS and relative QoS can always go into the same VLAN. When packets marked with absolute QoS (guaranteed traffic) arrive at the access node or access edge node they will always be forwarded. Packets with relative QoS (non-guaranteed traffic) are forwarded depending on the currently available (spare) rate. Fair (random) packet discard mechanisms make sure that all non-guaranteed packet flows are affected equally.

#### 4.1.6 Device Control

Device control is very important. After connection acceptance, the network edges must use per-flow policing based on the negotiated traffic parameters. Thus, the leaky-bucket parameters must be set accordingly, and the flow must be marked with the correct VLAN tag (p-bits). The intermediate nodes will handle frames according to the priority bits of the VLAN tag.

#### 4.1.7 N-RAF Mechanism

At first some description:

One pipe is summary of flows assigned to one Service Agent (SA). Every SA is identified by its S-VLAN ID and in case of pre-provisioned solution every pipe has only one S-VLAN ID and only one p-bit definition which identifies its QoS class

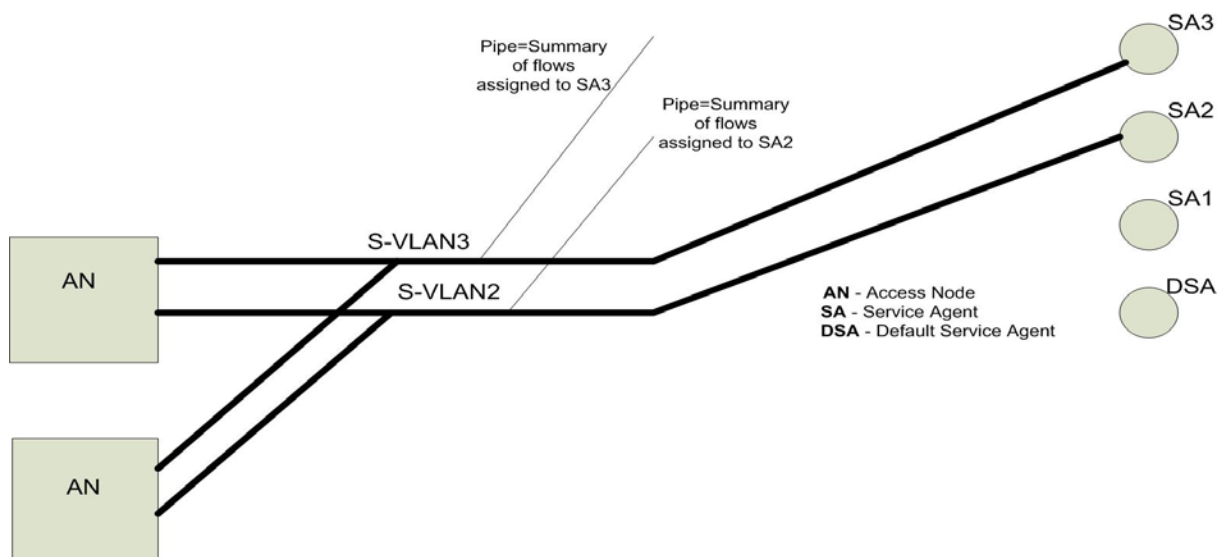


Figure 4-1 Pipe definition

First in the phase of SLA negotiation the SPANF sends SLA negotiation information and user information to the SAF. Next in the phase of SLS negotiation the SAF communicates with the

N-RAF and sends a resource allocation (link capacity) request with flow description and service related QoS description information to it.

The flow description parameters are: (Upstream/Downstream) token bucket size ( $\sigma$ ) and token rate ( $\rho$ ) which define the bandwidth in case of the leaky bucket mechanism.

The QoS description parameters are: delay, delay variation and packet loss. The suitable SA with its S-VLAN ID can be selected based on the service type and description parameters by the SAF.

The Admission Controller (AC) and the Bandwidth Broker (BB) are part of the N-RAF(See Figure 4-2). The AC receives this request with parameters and calculates the suitable bandwidth based on them. The suitable bandwidth value with Source (AN ID), Destination (EN ID) and S-VLAN ID are sent to the BB. The BB makes a decision by the current connectivity and link capacity (available bandwidth) information from the resource database (RDB).

Information in the RDB are: S-VLAN ID, Source (Access Node ID), Destination (Edge Node ID), priority bit-QoS class (every class has its own parameters like bit rate, error rate, delay, jitter and packet loss in the database), total resources (bandwidth), uncommitted resources. More than only one pipe can exist between the source and destination node and information are stored into the RDB.

Source parameters for the BB can be the Service Binding parameters, so the binding definition can be also the flow definition. Its output is the decision and if the request is accepted the suitable pipe or pipes list which information are also sent to the SAF.

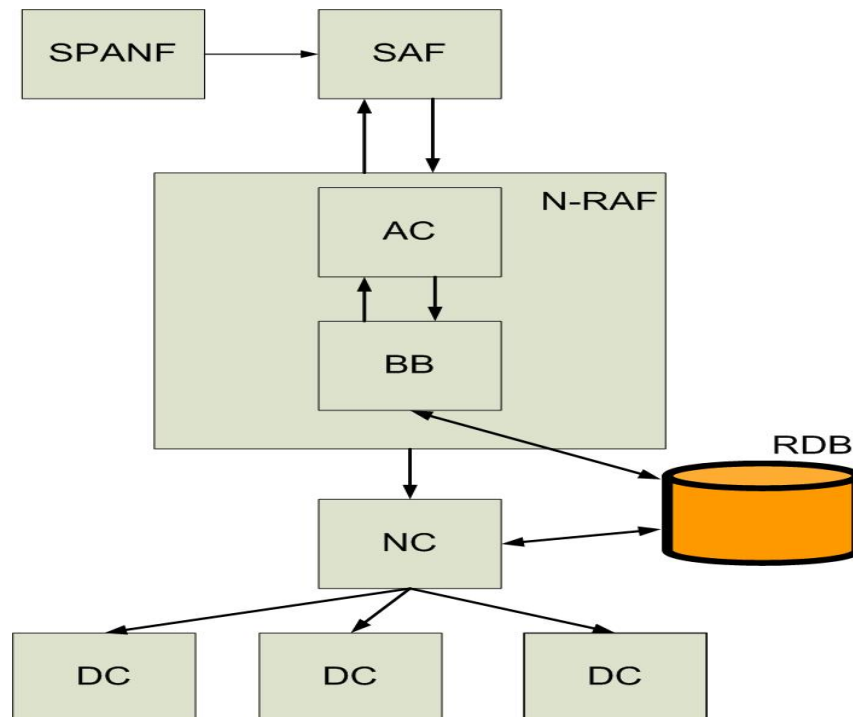


Figure 4-2 Flow diagram between the entities

The BB checks from the RDB if the requested resource is available or not. If the resource is available only for one pipe, the BB permits this resource allocation and the N-RAF sends this information back to the SAF and sends a bandwidth reservation request to the Network Controller (NC) for this pipe and put this information to the RDB. The NC also communicates with the Device Controller (DC) which has to set some leaky-bucket parameters (like rate limiting and traffic shaping) for devices on path for this pipe. So the reservation of resources can be used during the SLS negotiation phase but they are committed when the service flow is initiated.

If more than only one pipe can be used for this new flow, the BB permits the resource allocation and its output is a decision and the list of suitable pipes. The N-RAF does not make a selection which pipe will be used. The N-RAF sends this list to the SAF and it chooses which pipe will be used. After the pipe has been chosen the NC reserves this bandwidth for this pipe and put this information to the RDB and also communicates with the DC which has to set some leaky-bucket parameters (like rate limiting and traffic shaping) for devices on path for this pipe.

In case of acceptance the SLS can be created for the given service. The traffic entering the pipe will be delivered with this pre-defined QoS class as long as the reserved bandwidth is not exceeded.

If the resource allocation has been dropped or not fulfilled the SAF also will be notified and the SLS cannot be created for the given service.

In case of service deactivation the resource de-allocation is requested. The link capacity is released and resource information is modified in the RDB by the NC. When for the resource reservation communication are needed also between more domains the SAF have to communicate between each other, so the SAF-SAF interaction has to exist.

## 4.2 Service Subscription Management

This is like a “pair” of Service Definition (Section 3.1). This section describes how the user subscribes to a previously created service presented through the full lifecycle of a service from the user’s point of view.

This lifecycle is the following:

- Service definition
- Service subscription
- Service activation
- Service un-activation
- Service un-subscription
- Service locking
- Service un-definition

So this section also covers the user management related issues (e.g. databases of users as presented in Section 3.5, etc).

This section also describes how to represent and communicate the service-related parameters to the management system in a “computer-equipment friendly” format, while the Service Definition part gives a “user-friendly”, human-readable description of the same parameters.

## 4.2.1 SLA management

This section describes the content of the SLA, the negotiation parties and interfaces.

### 4.2.1.1 Content of the SLA

The SLA should contain all service parameters including the name, charging and term of the service. Note that by default some parameters are hidden from the users when selecting a service and subscribing to it, since they are “a priori defined” by the technology used by the network. These parameters are only visible at a detailed list of the service parameters. (For example, packet loss and method of bandwidth reservation.)

### 4.2.1.2 SLA negotiation parties

The SLA can be interpreted between different pairs of components in the networks. Since our work is closely related to the CADENUS based model, we investigate the pros and cons of these “SLA-pairs” in this model.

In case of CADENUS based model, we have the following reasonable versions from user- and network perspectives:

1. **User – SPANF**: The “content” of the SLA is typically agreed between the User and the SPANF based on the offer of the SPs communicated by the SAF to the SPANF.  
*Pros*: The Global Service Registry (GSR containing all available services in the network) is also maintained by the SPANF. One self-provisioning portal for all services.  
*Cons*: Large and centralized GSR (this is not a real disadvantage).
2. **User – SAF**: Not usual, since the user typically is not able to communicate with the SAF.  
*Pros*: Allows distributed GSR in the distinct SAFs.  
*Cons*: Direct communication is needed between the User and the SAF. Need for self-provisioning portal in each SAF.
3. **User – SP**: A possible way of agreeing on the “content” of the SLA. It could be used in case of content services, when the SP is not directly connected to an SAF, but through another SP.  
*Pros*: Possibility of embedded services (the availability of a service depends on the existence of another).  
*Cons*: Need additional negotiation phases when creating the SLA.
4. **SPANF – SAF**: The negotiation of the SLA is typically done between them after the User has selected the “content” of the SLA.  
*Pros*: A service is typically selected at the SPANF. The SLA can be considered as an interface between the SPANF and the SAF. The SLA database also maintained by SAF.  
*Cons*: Nothing.
5. **SPANF – SP**: Not usual, since the SLA has to be passed to the SAF in the end.  
*Pros*: Nothing.  
*Cons*: Need additional negotiation phases when creating the SLA.

So it can be seen, that the simplest and recommended way of agreeing and negotiating the SLA is the following:

- On the “user side”: the SLA is agreed between the User and the SPANF.
- On the “network side”: the SLA is negotiated between the SPANF and the SAF.

#### 4.2.1.3 SLA negotiation interfaces

The user communicates with the SPANF through the Provisioning Interface, through which a user can generate and agree upon a service contract (“content” of the SLA). See Section 3.6.1 for details.

The practical negotiation of the SLA is done through the Service Negotiation Interface. This interface represents a bridge between the users and the SPs and it is created between the SPANF and the SAF. See Section 3.6.2 for details.

### 4.2.2 SLS Management

This section describes the content of the SLS, the negotiation parties and interfaces.

#### 4.2.2.1 Content of the SLS

The SLS should contain the detailed list of the service parameters to be met, such as QoS, mobility, security, etc. The main focus and interest of this report is the QoS aspect of SLS, while mobility and security issues can be investigation areas for MUSE Phase II.

#### 4.2.2.2 SLS negotiation parties

The SLS can be interpreted between different pairs of components in the networks. Since our work is closely related to the CADENUS model, we investigate the pros and cons of these “SLS-pairs” in this model.

In case of CADENUS model, we have the following reasonable versions:

1. **SAF – N-RAF**: The negotiation of the SLS is typically done between them.  
*Pros*: The SLA database is typically maintained by the SAF, the SLS database is typically maintained by the N-RAF. So the SLS can be considered as an interface between SAF and N-RAF.  
*Cons*: Nothing.
2. **SPANF – N-RAF**: Not usual, since the SPANF typically is not able to communicate with the N-RAF.  
*Pros*: The SPANF can handle both the SLA and the SLS.  
*Cons*: The SPANF needs almost all functionalities of the SAF in order to create the SLS resulting unmanageable and unsynchronised SLA databases (all SPANFs have to store the SLA databases of all SAFs).

So it can be seen, that the simplest and recommended way of negotiating the SLS is that the SLS is negotiated between the SAF and the N-RAF.

#### 4.2.2.3 SLS negotiation interfaces

The negotiation of the SLS is done through the resource negotiation interface. This interface is used to exchange QoS resource reservation information between the SAF and the N-RAF. See Section 3.6.3 for details.

This interface is identical to the Rq reference point described in the TISPAN RACS document.

### 4.2.3 SLS and SLA Mapping

This section describes the connection of the SLA and the SLS defined for the “same service” in the network. The description is focused on the place of the mapping, the used interfaces and the storage of the corresponding databases.

#### 4.2.3.1 SLA-SLS Mapping Parties

In case of CADENUS model, there are two possible places, where the mapping of SLA to SLS can be done: in the SAF (where the SLA database is stored and maintained) or in the N-RAF (where the SLS database is stored and maintained). Both alternatives have their pros and cons as follows:

- **Mapping in the SAF:** the SAF is responsible for the mapping. The SAF requests parameter information (network setup) from the corresponding N-RAF(s) based on the content of the SLA. Thus the SLS-SLA mapping is a part of the SLS negotiation phase.  
*Pros:* The SAF can communicate with different N-RAFs based on the SAF’s quasi independence of the network technology (centralized view), so there is the possibility to support Fixed Mobile Convergence (FMC) and nomadism.  
*Cons:* Administrative overhead on the SAF. The exact knowledge of the network status is available only in the N-RAF. Multiple rounds of the negotiation are needed.
- **Mapping in the N-RAF:** the N-RAF is responsible for the mapping. The N-RAF gets the SLA information from the SAF and the N-RAF is able to give prompt answer with the SLS parameters conform to the SLA.  
*Pros:* The SLS negotiation can be done in one request-response round.  
*Cons:* In case of FMC and nomadism it results administrative overhead to the whole network because of the distributed coordination of the SLS mapping (since the SAF and the old and the new N-RAF are also involved in the negotiation).

In both solutions, the mapping of SLA and SLS requires some “extra” functionality, however, the centralized view seems to require less complex OAM system. Moreover, the centralized view seems to be more efficient when considering FMC and nomadism. So the recommended place of the SLA-SLS mapping is in the SAF. Furthermore, the sharing of SLA and SLS information within a management system or between different management system is also simpler in the centralized view (the place of the mapping is “a priori given” by the SLA).

#### 4.2.3.2 Used Interface

The required information exchange in order to provide the SLA-SLS mapping is obviously done through the Resource Negotiation Interface as in case of SLS negotiation in the CADENUS model or through the Rq reference point in the TISPAN model.

### 4.2.4 SLS and Policy Mapping

According to our view, as SLS is sent to the N-RAF via the RNI, the N-RAF has the responsibility to create the service related policy based on the SLS. Then the service related policy is sent to the NC via the NCI. The NC translates this information into network specific policies sent to the DCs.

### 4.2.5 Detailed list of service subscription related service parameters

We describe which parameters are visible and which are modifiable in each level.

According to Section 3.1, the service parameters can be splitted into two groups:

- *Constant, unvariable parameters.*
- *Modifiable and selectable (both mandatory and optional) parameters.*

In Table 4-1 those parameters are listed, which are technology specific constants.

Service parameters	Sub-parameters	SLA		SLS
		Normal user	Advanced user	
Addressing & numbering	Type of addressing & numbering	Phone number or Alias or IP address or Domain name	Phone number or Alias or IP address or Domain name	Phone number or Alias or IP address or Domain name
Security	Authentication	Port or ID + Password or Hardware key	Port or ID + Password or Hardware key	Port or ID + Password or Hardware key
	Monitoring	Not visible	Trace or Interception	Trace or Interception
	Coding & decoding	Not visible	Codec type	Codec type
Control	Way of parameter selection	Select from a list indirectly	Select from a list indirectly or directly	Set directly
	Remote control	Allowed or not (in correspondence with the level of the authentication)	Allowed or not (in correspondence with the level of the authentication)	Allowed or not (in correspondence with the level of the authentication)
Default connection	Default AEN	Address	Address	Address
	Default SA	Address	Address	Address
	Internal / external	Not visible	Not visible	Indicator
Booking	Bandwidth reservation	Not visible	Not visible	Reservation in advance Dynamic reservation

Table 4-1 Constant service parameters related to service subscription

The following service parameters are modifiable by the user or have to be given before the subscription. For example, in case of IPTV the user can modify his TV portfolio (list of TV-channels available for him) and in case of peer-to-peer connection the user has to give the required speed of the connection in advance. The modifiable parameters usually have many sub-parameters, so the *traffic-, the throughput characteristics* and *the Quality of Service* parameters are introduced in separated tables.

First the sub-parameters related to the *Traffic characteristics* are listed in Table 4-2.

Service parameter	Sub-parameters	SLA		SLS
		Normal user	Advanced user	
Traffic characteristics	Real-time	Not visible	Yes or No	Indicator
	Interactive	Not visible	Yes or No	Indicator
	Unidirectional	Not visible	Text, only one of them can be selected	Indicator, only one of them can be selected
	Bidirectional			
	Asymmetrical	Not visible	Text, only one of them can be selected	Indicator, only one of them can be selected
	Symmetrical			
	Point-to-point	Not visible	Text, only one of them can be selected	Indicator, only one of them can be selected
	Point-to-multipoint			
	Unicast	Not visible	Text, only one of them can be selected	Indicator, only one of them can be selected
	Multicast			
	Broadcast			
	Concast			
	Anycast			

Table 4-2 Modifiable service parameters I.: Traffic characteristics

Note that most of the normal users do not care about the *Traffic characteristics* parameters, instead they use the default setup given by the system after choosing the given service based on the constant service parameters (practically based on the description).

In Table 4-3 the *Throughput characteristic* related sub-parameters are listed. In case of normal user, they are connected to a service grade definition like bronze, silver, gold, premium, etc.

Service parameter	Sub-parameters	SLA		SLS
		Normal user	Advanced user	
Throughput characteristics	Burst traffic	Not visible	Text, only one of them can be selected	Indicator, only one of them can be selected
	Continuous traffic			
	Number of channels	Number indirectly from a list (e.g. bronze = 1, silver = 2, gold = 4 parallel TV channels)	Number from a list	Number
	Constant bit rate	Number indirectly from a list for both directions	Number from a list for both directions	Number for both directions
	Variable bit rate	(e.g. bronze = 4 Mb/s, silver = 8 Mb/s, gold = 16 Mb/s internet access)		
	Maximum bit rate			

Table 4-3 Modifiable service parameters II.: Throughput characteristics

In Table 4-4 the *Quality of Service* related sub-parameters are listed. In case of normal user, most of them are hidden, however, the availability is connected to a similar service grade definition as above.

Service parameter	Sub-parameters	SLA		SLS
		Normal user	Advanced user	
Quality of Service	Bit error rate	Not visible	Not visible	Number
	Packet loss	Not visible	Not visible	Number
	Delay	Not visible	Number from a list	Number
	Jitter	Not visible	Number from a list	Number
	Availability	Text indirectly from a list (e.g. bronze = 1 day -, silver = 1 hour -, gold = 5 minute outage per year)	Number from a list (e.g. 99.999)	Number
	Accessibility	Not visible	Number from a list	Number
	Retainability	Not visible	Number from a list	Number

Table 4-4 Modifiable service parameters III.: Quality of Service

## 5 SERVICE ASSURANCE FRAMEWORK

This chapter describes a framework for Service Assurance (SAS) functions, methods and activities. We use a generalized meaning of Service Assurance, namely: it covers all the functions that help *assuring* fluent *services* over the managed network. SAS systems are not mandatory when deploying a service, since in a perfect world services run without problems from the first moment. Once we begin to suspect that the world is not perfect, we realize the necessity of SAS.

The proposed SAS framework covers Connectivity Fault Management (CFM), Performance Monitoring (PM) and SLA Verification, and processes the output *event notifications* of these (together with some other events detailed later).

This chapter is organized as follows. After the short overview of the SAS framework we describe the main elements in sub-chapters. The following overview describes the framework from the event processing point of view (from the Fault Management (FM) prospective). The sub-chapters describe how the events (CFM, PM, SLA Verification, etc.) get generated and fed into this FM-flow, then detail the elements of the FM-system.

Before describing the SAS-framework, we should make clear the difference between *event* and *alarm* notifications. *Events* are generated by different parts of the managed system, notifying about status changes minor, major and critical errors occurred at a given object. These events are sometimes alarming, but most of the time harmless for the overall system's point of view. *Alarm* notifications, however, are clear, trouble ticket like objects, which in all cases need to be acted upon. There is always a fault behind an alarm notification.

Figure 5-1 depicts the suggested event processing and alarm handling model. This model is a detailed and wide-ranging description of the N-RAF functions. The SLS Database can be found in Section 3.5.3.

The functions of the main processing elements are the following:

- **Event Notification Collector** – The FM-event sources are all addressing their notifications to this module. It collects the events sent in different formats and reshape them to a standard notification format. Events will be stored in this standard format into the *Event Notification Database* and sent to the *Event Processing* module also. The event-generator entities are:
  - *Connectivity Fault Management* – The CFM sends Alarm Indication Signals on connectivity issues. The framework is described in section 5.1.
  - *Unsolicited Performance Monitoring* – Performance Monitoring measurements are carried out at each node in an unsolicited manner (periodic checks without any external trigger). These are described in section 5.2.
  - *SLA Verification* – SLA and SLS verification procedures are triggered according to section 5.3. The metrics and checking routines are mostly identical to the ones used in *Unsolicited Performance Monitoring*, the initiation of these checks, however, is different. The *SLA Verification* framework includes a knowledge base about the SLA and SLS metrics to be met at different service levels. Solicited performance measurement routines are carried out based on this information.
  - *Syslog* – Each node in the managed service area generates *syslog* messages. These should arrive to the *Event Notification Collector*, also.
  - *Security* – Events generated by the Security Management system should be used for fault management purposes.
  - *Other SNMP Traps* – There are various other types of SNMP traps that should arrive as event notifications, but do not fall into the above categories. Such traps arrive for example from the Bridge nodes, notifying about the occurrences of buffer overload or exceeded capacity-limits.
  - *User* – The operator should log event (rather: alarm) notifications as well.
  - *Data Miner* - The source module of these events executes outlier detection and trend analysis algorithms on the complete set of event notifications arriving from other sources.
- **Trend Analysis and Outlier Detection** – This module takes its input from the *Event Notification Database*. It continuously runs *Trend Analysis* algorithms and detects *Outliers* that do not fit into patterns on the data set and generates alarming events targeted to the *Event Notification Collector*.

**Event Preprocessing** – It consists of a *Correlator* and a *Filter* module. The *Correlator* is a rule based event correlation module. If a correlation rule is matched, a new event notification is generated. This sub-module passes all event notifications (either “correlated” or simple) to the *Filter* sub-module.

- **Alarm Notification Presentation** – The output of Event Preprocessing are the *alarm* notifications. These will be presented to the *Network Management Center* so the operator will see the distinguished, active alarms. Each alarm notification will trigger a new *Root Cause Analysis* entity.

- **Root Cause Analysis** – Based on the descriptive parameters of the alarm notification, this module tries to find the root cause of the problem. It fetches data from the *Event Notification Database* and the *Topology Database* and initiates active checks based on this information. The RCA framework is detailed in section 5.4.
- **Advice and Modification** – The RCA output is a fault description. The *Advisor Module* compares the parameters of this description with its knowledge base, and provides a suggestion for corrective actions. It depends on the operator’s policy if he/she allows the system to make corrective actions by itself. The corrective actions certified to be done by the system automatically will be passed to the appropriate *Modification Agent*. The other actions and suggestions will appear at the *Network Management Center*, so the operator can carry out the necessary steps. This module is detailed in section 5.5.
- **Event Notification Database** – It stores all event notifications sent towards the management system. The notification description is standardized (regardless of the source) – it must contain all information to be used by the FM system (alarm ID, priority, source, type, relevant connection addresses, short description where possible, other distinguishing parameters and values).

**Topology Database** – There are at least three important pieces of information stored here about each node. These are: Node address, Node type (functionality), List of connecting nodes (first hops, subnet addresses

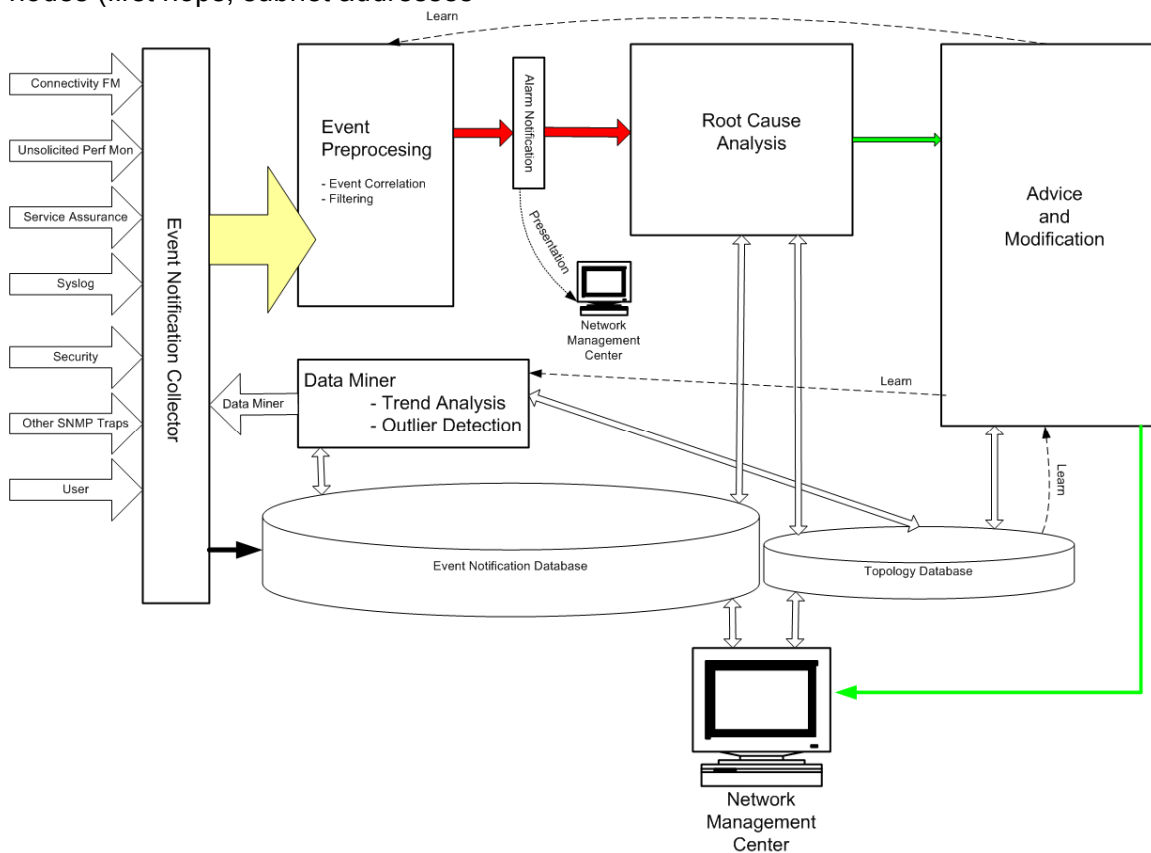


Figure 5-1 - Service Assurance Framework

As Figure 5-1 suggests, events arrive from different sources. The Event Notification Collector maps these to a standard format, then sends them to Event Preprocessing and stores them into the Database. The Data Miner works on the stored events (seeing more historical events), and generates a new event when necessary. Once the events get correlated and filtered, only those *alarms* get presented to the NMC, which should be acted upon. The Root Cause Analysis of these starts immediately. When the root cause is found, a fault description gets forwarded to the Advisor Module. This should be able to decide whether the corrective actions should be taken automatically, or only the Advice (and the log about the tests carried out during RCA) should be forwarded to the human operator. The operator then decides what to do and is also able to check if the automatic corrections taken were appropriate. The model includes learning mechanisms, which are also shown by Figure 5-1.

## 5.1 Connectivity Fault Management

The Connectivity Fault Management (CFM) in MUSE should rely on existing Ethernet CFM standards. The Service Assurance framework uses the functions of the standard-based CFM. These functions are everywhere in the architecture where Ethernet exists.

There are different technologies from different standards bodies for carrying Ethernet services:

- IETF: Ethernet over MPLS, Ethernet over L2TPv3, VPLS meshes
- ITU-T: Ethernet over SDH
- Ethernet over cable modems
- IEEE 802.1 and 802.3

The only thing that is constant across all technologies is the Ethernet frames, therefore CFM is Ethernet frames, not MPLS, ATM, or SONET frames, packets, cells, or physical layer control information.

Connectivity Fault Management refers to the ability of a network to monitor the health of a service delivered to customers as opposed to just links or individual bridges. Specifically, CFM scans the operation of the bridges and links, and the connectivity across a multi-operator, multi-provider network. The goal of Ethernet CFM is to monitor an Ethernet network that may be comprised of one or more Service Instances. A Service Instance is a LAN that may be made up by various means, including a VLAN, a concatenation of VLANs, etc. Since no operator has complete coverage of a large region, a Service Instance would span the provider network covering one or more operators.

There are two key elements of CFM: Maintenance End Points (MEPs) and Maintenance Intermediate Points (MIPs). MEPs and MIPs are new software (or potentially hardware) entities created within a bridge for CFM only. MIPs and MEPs can be implemented per bridge or per port. MEPs initiate CFM messages and respond to CFM messages. MIPs passively receive these messages and respond back to the originating MEP.

A Maintenance Association (MA) is a logical connection between any two MEPs. A Maintenance Domain consists of one or more MAs at the same level. There are eight levels defined in CFM. Level [7] is for ETH Section monitoring, levels [6,5,4,3,2,1] are for the EVC segment and level [0] is for the EVC path. As a default, levels [6, 5] are for the network operator, levels [4, 3] are for the service provider, and levels [2, 1, 0] are for the customer.

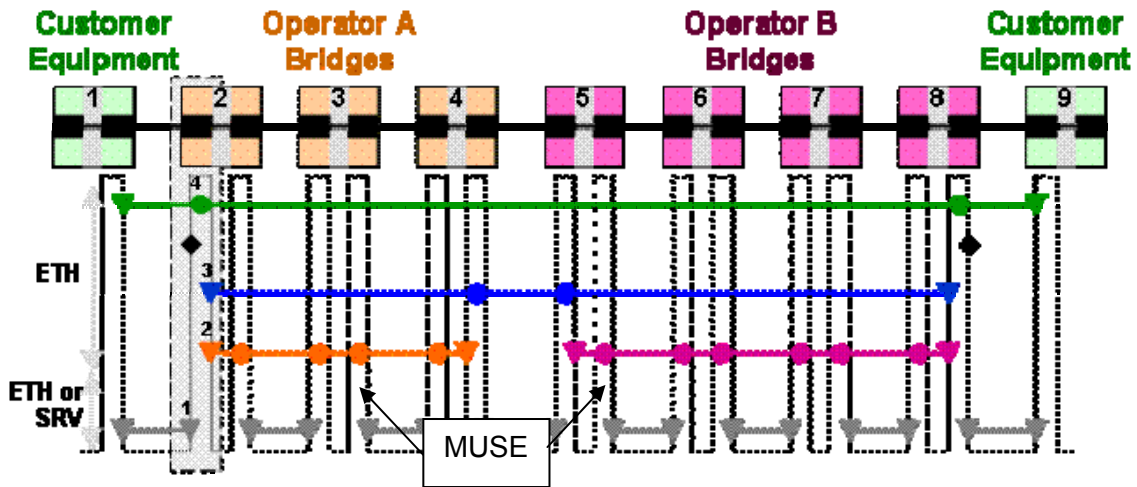


Figure 5-2 - Relationship between MEPs/MIPs and Port Status

Figure 5-2 shows the cross-section of the horizontal CFM plane. The dotted lines indicate the path along which a CFM frame would travel as it proceeds from one customer equipment to the other. Triangles indicate MEPs, circles indicate MIPs at different levels shown by different colors.

The MUSE architecture does not necessarily cover the end-to-end service, thus the CFM should be considered at the operator level, as indicated on Figure 5-2. In the MUSE architecture the MEPs should reside in the Access and the Edge node, while all bridge ports in the network are MIPs.

Mechanisms supported by 802.1ag include Connectivity Check (CC), Loopback, Link trace and Alarm Indication Signal (AIS). CFM allows for end-to-end fault management that is generally reactive (through Loopback and Link trace messages) and connectivity verification that is proactive (through Connectivity Check messages). Additional mechanisms supported in [Ref 16] include Remote Defect Indication (RDI), Loss Measurement (LM) and Automatic Protection Switching (APS).

#### CFM Packet Types

- **Connectivity Check:** Connectivity Check (CC) messages are periodic hello messages multicast by a MEP within the maintenance domain. All MIPs and MEPs in that domain will receive it but will not respond to it.
- **Link-trace:** The Link trace message is used by one MEP to trace the path to another MEP or MIP in the same domain.

**Loopback:** A Maintenance Association (MA) is a logical connection between any two MEPs. A Loopback message helps a MEP identify the precise fault location along a given MA.

- **Alarm Indication Signal (AIS):** The Maintenance Entity sends it when detect failure in the connection.

The periodic Connectivity Check (CC) messages should flow between MEPs in the MUSE network. When connectivity loss is detected, the loss of connectivity is signaled to the Alarm framework by AIS signals.

The Service Assurance framework uses the AIS signals for failure notification, and the loopback/linktrace functions provided by CFM for alarm localization and verification.

---

There are two draft standards about CFM:

- IEEE P802.1ag Connectivity Fault Management [Ref 15]
- ITU-T Study Group 13 Question 5 Y.17ethoam [Ref 16]

## 5.2 Performance Monitoring

In our Service Assurance Framework, dedicated OAM functionalities notify the network management about defects occurred in the network. The localisation and the diagnosis of the problem is the task of the Root Cause Analysis (RCA) module, however, the Performance Monitoring (PM) module has the dedicated task to send Notification Events about the defects. For the sake of the cause, PM carries out periodical unsolicited measurements in order to provide data on the most important performance descriptive parameters of the network. Whenever the value of a parameter reaches a given threshold, this module sends a message to the Event Notification Collector (ENC). Since the connectivity status assessment should not be dependent on the dynamic behaviour of the customer traffic, this type of defects are analysed by the Connectivity Fault Management (CFM), see Section 5.1 for details. Furthermore, the "down" time of the service should be able to be recorded for performance and availability measurements by the CFM.

### 5.2.1 Measurement objectives

In order to get an adequate view of the network status, all links of the network should be monitored. On the one hand, the switching is deterministic inside the access network (or partly covered by MUSE) meaning that the Resource Management determines the routes used by the traffic of the users. On the other hand, the user traffic can enter or leave the network only at the Access Node (AN) or at the Edge Node (EN). Therefore, it is enough to carry out the network performance related measurements between the AN and the EN.

Since different minimal requirements are specified for the different service classes, the measurements have to be done for each service class between each AN-EN node pair. Moreover, if VLAN-based trees are used for traffic engineering proposes, the measurements have to be applied for each possible VLAN-based tree containing a particular AN.

The Metro Ethernet Forum [Ref 8,Ref 9] specifies Ethernet service attributes, based on which the following performance parameters are proposed to be measured in the network. These parameters are Frame Loss Ratio (FLR), Frame Delay and Frame Delay Variation (jitter). However, this list can be completed with Bit Error Rate (BER). However, concerning OSI layer 3 functions, CISCO proposes similar parameters to measure [Ref 12].

All of these performance parameters can be measured at the same time on a synthetic traffic injected into the network. This synthetic measurement traffic does not disturb the traffic of the users and can be handled separately to provide easier administration for the network management. (For example, special locally administered virtual MAC address is allocated to the measurement traffic.)

In order to provide realistic measurement of the above parameters, the network should met the following assumptions:

- Adequate Policing should be applied in order to allow weighted fair queuing in the AN and the EN.
- Timestamp can be added to the measurement traffic.
- Synchronisation of the clocks if one-way delay-related measurements are needed.

### 5.2.2 The measurement process

The synthetic measurement traffic should be inserted into the network at the Edge Nodes. Based on the Topology Database and the configuration information (VLAN assignment) provided by the Central Network Resource Manager (CNRM, see Section 3.3 for details), all AN-EN route pairs can be calculated (concerning all service class and available VLAN-based trees) and the measurement can be done on all of these routes. The framework of the measurement process can be summarized as follows:

1. Select the next route to be monitored, which route is determined by the end-points (AN and EN), service class and the VLAN.
2. Insert the measurement traffic at EN, which is practically a test frame:
  - a. Sets the source address to address of the EN and the destination to the address of the AN.
  - b. Inserts a timestamp into the test frame.
  - c. Sends the test frame to the AN.
3. The AN receives the test frame and swaps the destination and the source address of the test frame.
4. (If the clocks are synchronised in the network, insert a timestamp again.)
5. The AN sends back the test frame to the EN.
6. The EN processes the test frame (timestamps and bits of the frame) and sends it to the Performance Monitoring module.

Since the tests are done periodically on all possible AN-EN route pair, all four parameters can be measured with the test frame. Here is the short summary of how to calculate them for each service class.

1. *Frame Loss Ratio* calculation based on measurements statistics:  $FLR = (S-R)/R$ , where S is the number of sent frames per time unit and R is the number of received frames per time unit in the EN.
2. *Frame Delay* calculation based on the timestamps:
  - a. Round-trip delay can be calculated by the difference of the timestamp inserted into the frame by the EN, when it sends and receives the particular test frame.
  - b. One-way delay can be calculated if AN can insert “synchronised” timestamp into the test frame similarly to the round-trip delay.
3. *Frame Delay Variation* calculation: it is simply the difference between two consecutive delay values for the same AN-EN relation.
4. *Bit error rate* calculation: since the content of the test frames are known, it can be easily calculated by comparing the content (bits) of the sent and the received frames. The BER is the number of different bits per a given time unit.

### 5.2.3 Performance measurement functions

The performance measurement functions monitor particular AN-EN pairs at given service class and VLAN. In case of failure, the PM sends an error message only with the suspicious parameter or can send all information up on request.

### 5.3 SLA verification

As stated before, the network management is informed about the defects by dedicated OAM functionalities/modules. The SLA Verification (SLAV) module has to monitor all parameters agreed in the SLA, however, the most of them is provided by other modules. The SLA verification is part of the N-RAF functionality.

The connectivity related parameters are requested from the Connectivity Fault Management (CFM) and the network performance related parameters are from the Performance Monitoring (PM). Also in that case, the exact localisation and the diagnosis of the problem is the task of the Root Cause Analysis (RCA) module, however, the SLAV module has the dedicated task to send Notification Events about the SLA violations to the Event Notification Collector (ENC). The SLAV mostly carries out unsolicited measurements to provide data on the active SLA types. However, it is very important to check the SLA of particular users, where checking can be triggered by the Basic Check Routines or by a systematic review of all active subscriptions in the network. In case of the most of the monitored parameters it is advantageous not to monitor individual user SLAs, but to monitor SLA groups, since these parameters describe the interior of the network and they are handled similarly as their routes are the same between the Access Nodes and the Edge Nodes. Therefore, it is enough to check the strictest SLA values on each route.

#### 5.3.1 Measurement objectives

The task of the SLA Verification module is to check whether the service parameters are in the agreed range or not. The service parameters are listed in Section 3.1 and in Section 4.2.5, however, here only those parameters will be monitored, which depend on the performance of the network. These parameters are connected to the Throughput characteristics and to the Quality of Service. On the other hand, these parameters can be grouped according to which module or network element monitors them:

- CFM: Availability, Accessibility and Retainability.
- PM: Frame Loss, Frame Delay, Frame Delay Variation (jitter) and Bit Error Rate.
- Access Node & Edge Node: Throughput at egress, Offered Load at ingress and Frame Rate at egress and ingress.

Some assumptions have to be met in order to provide realistic measurement of the above parameters. For assumptions related to CFM and PM please see Sections 5.1 and 5.2. An additional assumption connected to the SLAV is the ability to measure the amount of the incoming and outgoing traffic at the AN and the EN per each subscription. (Note that this is practically fulfilled, since it is also required by any billing system independently from MUSE.)

Due to the predefined routes between the ANs and ENs, it is not needed to monitor all SLA related parameters for all subscription individually. For a given AN-EN node pair, the parameters connected to CFM and PM can be measured per service class on each available route. However, the Throughput, the Offered Load and the Frame Rate have to be monitored individually.

The first mile and the network towards external content providers are out of the scope of this work, so only connectivity check and throughput check can be done for those parts of the network on the Ethernet level. However, if delay, jitter or bit error rate related measurements and verification are needed, then it should be done on the IP level for example by UDP and ICMP measurements as defined by Cisco’s white paper on IP SLAs [Ref 10, Ref 11].

### 5.3.2 The measurement process

According to the above, the most of the parameters can be measured in groups. So it is assumed that the SLAV module can build a list of the actual SLAs (gathered from the SLA database, see Section 3.5.3 for details). This list contains the SLAs in descending order based on their strengths for each service class per each possible route for a given AN-EN node pair. Let us show a short example highlighting which parameters are included and how the lists should be look like. (Note that the lists can be united in a single table as in Table 5-1)

SLA group between AN 23 and EN 3: Service Class = 2 <sup>nd</sup> priority, Route = VLAN 83						
Availability	Accessi- bility	Retainability	Frame Loss	Frame Delay	Frame Delay Variation	Bit Error Rate
99.999%	99.999%	99.999%	10E-9	200ms	10ms	10E-9
99.99%	99.99%	99.99%	10E-8	350ms	15ms	10E-8
99.9%	99.9%	99.9%	10E-7	500ms	20ms	10E-7

Table 5-1 An example table for group checking of SLAs

So the measurement process for the above parameters can be done continuously as follows:  
 Consider the next SLA group and call CFM and PM to provide the actual measured values for this group.  
 Compare the measured values with the expected values.  
 If there is violation, then notify the Event Notification Collector about which parameter is violated and to what extent.

The rest of the parameters have to be measured individually for all subscriptions. These parameters are Throughput, Offered Load and Frame Rate. If proper resources are available to exhaustive verification of all subscriptions, then the process is simply to compare all measured parameters with the ones declared in the SLAs. If only limited computing resources are available, then we propose to check them quasi randomly. This means that all subscriptions are checked in a larger timeframe (e.g. once in a minute) and the randomization provides more realistic check compared to a strict periodical check. If there is violation, then the SLAV notifies the Event Notification Collector about which parameter is violated and which subscription is concerned.

### 5.3.3 SLA verification functions

SLA verification can monitor SLA groups automatically with limited capabilities or check an individual SLA of a subscription up on request.

---

## 5.4 Root Cause Analysis

There are three major types of Root Cause Analysis (RCA) systems have appeared in the last decade of fault management:

- a. systems based on alarm correlation algorithms,
- b. methods using statistical facts,
- c. model based approaches.

The merge of these methods and approaches could also lead to useable and effective RCA systems. State of the art fault management systems, however, do not rely on passively collected event notifications only. At some points of the analysis, the RCA algorithm should be able to initiate active tests.

The two ultimate questions in these systems are: *when* to initiate a test and *how* that test should look like. This section proposes a complete FM model, addressing the issue of scheduling active tests, also.

*The proposed RCA-method: Model Based approach*

Model based RCA solutions model the network, their events and the connections between them. The model is flexible, so topology changes do not lead to RCA algorithm changes or rule changes. The network topology consists of Node names and IP addresses, which has first hops (also IP addresses) associated with it. Beside the network model, the alarm model exists in a template-manner, also. This means that the network entities are only referenced in the alarm model (as parameters) rather than being hard-coded in there.

The input of the proposed RCA model are alarm notifications, which were generated based on a high number of event notifications, through an extensive alarm correlation and filtering process. This propagates that the types of input alarms become limited.

There is a powerful evaluation method, which takes into account that

- a. input data for elementary checks should also be fetched from somewhere and
- b. elementary checks should be carried out simultaneously in order to get faster results.

This method - introduced in the following paragraphs - uses Petri nets to schedule the elementary checks.

A brief description on Petri nets from *en.wikipedia.org*:

“As such, a Petri net has place nodes, transition nodes, and directed arcs connecting places with transitions. At any one time during a Petri net's execution, each place can hold zero or more tokens. Unlike more traditional data processing systems that can process only a single stream of incoming tokens, Petri net transitions can consume tokens from multiple input places, act on them, and output tokens to multiple output places. Before acting on input tokens, a transition waits until the following two conditions are met:

- (i) a required number of tokens appears in every one of its input places, and
- (ii) the number of tokens in each of its output places falls below some threshold.

---

Transitions act on input tokens by a process known as *firing*. When a transition fires, it consumes the tokens from its input places, performs some processing task, and places a specified number of tokens into each of its output places. It does this atomically, namely in one single non-preemptible step. Since more than one transition on a net can be firing at any one time, Petri nets are well suited for modelling concurrent behaviour of a (geographically) distributed system.”

To apply this for RCA, transitions should be the *elementary checks*, whereas places should represent *input parameters* (e.g. IP and interface addresses) and *output parameters* (results). A token marking a *place* means that the data associated with that place is available. This way all those transitions (*elementary checks*) can *fire*, for which the input places are all “tokened”.

Each alarm notification should have an own Petri net associated with it. To avoid high number of Petri nets to be designed, a passive filtering and alarm-correlator module should be applied in the system. This module should merely feed critical alarm notification to the Petri net based module.

## 5.5 Advisor Module and Modification

After recognizing the core of failure, the “network reconfiguration” advisor module decides if the problem can be solved. There are many solutions, this section presents some general groups of algorithms.

*Giving advice:* The recent network information is stored in the topology database. This module gives one or more advice for incoming alarms, which are stored in event notification database, by this topology database. More different algorithms can be used for incoming alarms and giving advice. Each one is kind of optimizing algorithms for raise the network performance. The choice can be made by preliminary preferences and/or adaptive learning behaviour.

Two types of advice can exist in the system:

*Network reconfiguration:* This type changes the network logical and/or physical structure. Advice for alternate physical configuration should be increasing and/or reorganizing network capacity. Optimizing routing algorithms executed on hypothetical full mesh network could offer relevant solutions.

*Network element reconfiguration:*

This type of advice concerns network elements. This group should include modifying queuing models, algorithms of processor time sharing etc.

An advice should have priority, which is inherited from alarm type and has three dimensions. These are the seriousness of changing, the urgency and the expected performance augmentation of recommended action(s).

The network management center is noticed of all pieces of advice, with their priorities, root alarms and occurred direct actions. The state of network preceding the direct actions must be logged for able to reverse their effect.

---

After classifying the possible logical reconfiguration the direct action is done if the priority is sufficient.

## 5.6 Trend analysis

The Data Miner is looking for trends and detects outliers on the complete set of events from the Event Notification Database. This subchapter describes some general methods.

### Outlier detection in time series data

Intuitively, outliers can be defined as given by Hawkins [Ref 6]. Outliers are outlying samples found in time series data which are useless for forecasting. These pieces of data are produced by random errors (e.g. route failures, DoS attacks, misconfiguration, etc.).

Outliers can have significant impact on the estimates of the model parameters of the time series data. The existing approaches to outlier detection can be classified into the following five categories:

- Distribution-based approach
- Depth-based approach
- Clustering approach
- Distance-based approach
- Density-based approach

Proposal algorithm is based on the L.O.C.I (LOcal Correlation Integral) algorithm [Ref 7].

Proposal method is the Autoregressive Integrated Moving Average Model.

Trends and outliers can be detected on traffic, local and global network performance data, such as bandwidth, QoS classes, SLA, SLS support, congestion, network element efficiency etc.

The goal is to predict failures according to threatening trends and to signal single and/or periodical outliers in network data. This feature gives a useful tool for more optimal and more efficient network functionality.

The predicted failures and detected outliers represented for following modules as predefined alarms. The model treats signals from statistical module like alarms from network.

## 5.7 Adaptation and learning capabilities

The presented model has adaptation and learning capabilities.

The learning and adaptation feedback information is shown on Figure 5-1.

The adaptation is important when modification has been done in the network. Besides the database update, all modules depending on modified data must refresh their inputs according to the new information. For example, the trend analysis and outlier detection must restart the observation the filters must restart their counters etc.

The learning capability of different modules can enhance the overall performance of the network by enhancing the reaction of different modules to typical error scenarios.