SLA Framework Development for Content Aware Networks Resource Provisioning

Cristian Cernat, Eugen Borcoci, Vlad Poenaru
Telecommunication Department
University POLITEHNICA of Bucharest
Bucharest, Romania
cristian.cernat@elcom.pub.ro, eugen.borcoci@elcom.pub.ro, vlad.poenaru@elcom.pub.ro

***Abstract* — This paper develops a Service Level Agreement (SLA) framework as a part of a complex networked media multi-domain eco-system, aiming to deliver multimedia Quality of Services (QoS) enabled services over multiple domain network infrastructures. It continues a previous work which defined the management architecture of a networked media oriented system - based on new Virtual Content Aware Networks (VCAN) concepts. The contribution of this work, beyond traditional SLA usage, consists in fully specifying and then implementing the dynamic SLA-based management for VCAN resources provisioning in a multi-provider, multi-domain network environment.**

Keywords — Content-Aware Networking; Service Level Agreement, Multi-domain; Management; Resource provisioning; Future Internet.

I. INTRODUCTION

A strong orientation towards content/information is expected for the current and Future Internet (FI), estimating that content distribution will cover approximately 90% of the total traffic in 2015 [1-4]. Among new architectural concepts proposed, there are Content-Awareness at Network layer (CAN) and Network-Awareness at Applications layers (NAA). This approach can create a *cross-layer optimization loop* between the transport, applications and services, which did not exist in traditional layered architectures and also better serves the content/information centric trends, [4].

The European FP7 ICT research project, "Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments", ALICANTE, [5], adopted the NAA/CAN approach. It defined, designed and currently is implementing this complex multi-domain Media Ecosystem. The work described here is a continuation of the research work associated with a part of this project. That is why, the complete system description and main system level architectural and design decisions cannot be detailed here; references are indicated to help the reader to get a more complete view on the system.

Several cooperating environments and business actors are defined in [5]: *User Environment* (UE), containing the End-Users; *Service Environment* (SE), containing Service Providers (SP) and Content Providers (CP); *Network Environment* (NE), comprising a new business entity called CAN Provider (CANP) and the traditional Network Providers (NP). An NP manages the network elements, in the traditional way, at IP level. The "environment" is a generic grouping of functions working for a common goal and which possibly vertically span one or more several architectural (sub-) layers.

This paper further specifies and develops a SLA – based framework to create and dynamically provision VCANs. The VCANs in ALICANTE approach can be seen as a light virtualization solution, i.e. parallel multi-domain logical Data planes, each one dedicated to certain classes of services and types of media flows. This approach is similar to that presented in [9] but enriched here with content awareness and QoS capabilities. The CAN Provider creates VCANs at request of the Service Provider which negotiate SLAs with CANP. Inside each VCANs the QoS assurance is performed by several mechanisms: static and dynamic provisioning, transport service differentiation and media flow dynamic adaptation.

The paper continues the work presented in other related papers such as [13-15] with focus on SLA framework as a part of the service and resource management. In addition to the previous work, *this paper develops the complete design, implementation and validation of the solution.* The paper structure outline follows. Section 2 presents related work. Section 3 provides a short overview of the system architecture. Section 4 develops the full SLA framework. Section 5 presents samples of implementation solutions. Conclusions and perspectives are outlined in Section 6.

II. RELATED WORK

Complex communication systems involve several business entities playing providers and customers roles. Usually the relationships between such entities are described by SLAs, [5-10].The technical part of the SLA is frequently named *Service Level Specification (SLS)*. While SLA have been proposed in many studies, their usage especially in a dynamic context is still an open research topic.

In MESCAL project, [6] the SLS concepts are extended to cover multiple-domain networks. It is defined a *Service Provider* actor dealing with higher level services and *IP Network Providers* offering connectivity services. Additionally, the concept of Meta QoS class is introduced in

 ISBN: 978-1-61208-279-0

[7], as a set of well known identifiers and parameter ranges for a few number of service classes, covering all significant application flows types, aiming to simplify the inter-domain signaling related to QoS composition. In [8] a system for media distribution over multiple heterogeneous network domains is proposed. The business model includes several actors: *Service Provider (SP), Network Provider (NP), Content Consumer (CC), Content Provider (CP), Access Network Provider (ANP).* Several types of SLA/SLSs exist customized to fulfill the needs of media related applications and services. However, all three solutions presented above consider only a single logical network infrastructure serving different traffic flows.

A new concept of *Parallel Internets (PI)* is proposed in [9], to enable end-to-end (E2E) service differentiation across multiple administrative domains, based on logical slicing of the Internet. The PIs can coexist, as logical networks composed of interconnected, per-domain, Network Planes (NPl). The actor types are Service Provider (SP), IP network provider (INP) and customers. INPs offer IP connectivity services and SPs offer high level services to customers. Several types of SLAs are defined, and also INP interconnection agreements (NIAs). However, no content awareness concept is present in [9].

The FP7 project COMET *"COntent Mediator architecture for content-aware nETworks",* [10] provides a *unified* interface for content access whatever the content characteristics are: *temporal nature, physical location, interactivity requirements*, etc. It enables the most appropriate E2E transport by mapping the content (according to its requirements and user preferences) to the appropriate network resources. The objective is to get the best quality of experience (QoE) for end users; it supports unicast, anycast and multicast. A Content Mediation plane is introduced between ISPs and content servers, combining content resolution and access: locating content according to delivery requirements (content mediation); delivering it using the most suitable resources (network mediation). SLAs are defined between the business entities.

In ALICANTE, [5], [12], a similar concept to parallel virtual planes [9] is adopted, but modified and enriched with *content awareness*. There will be (generally) a one-to-one mapping between a network data plane and a VCAN - where each VCAN is customized for a given class of services and type of media flows. The COMET and ALICANTE have overlapping scopes. However, the COMET business model is only partially sufficient for ALICANTE needs: no powerful user and service environment exists; service environment functions are embedded in ISPs; no CAN Provider exists. COMET does not fully consider the cooperation between network overlay and network resources, but is focused mainly on mediation activities. There is no complete chain of services management. ALICANTE considers additionally the concept of home network and its associated Home Box and exploits CAN concepts and optimization loop between applications/services and network.

In a recent work [11], a content centric solution is proposed. The CURLING, "*Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services*", aims to media content distribution at massive scale. It has a holistic approach, (content publication, resolution and, delivery) and provides to Content Providers and customers high flexibility, to express their location preferences when publishing and requesting content, respectively - through *scoping* and *filtering* functions. Business relationships are defined between ISPs, including local ISP policies, and specific CP and customer preferences. ALICANTE is complementary to CURLING; it is less content centric in the control plane, but more powerful in assuring efficient media flow QoS- enabled transport, based on content awareness. The SLA framework is very flexible allowing dynamic creation, modifications and termination of VCANs.

III. Content Aware networks –system Architecture

The ALICANTE concepts and architecture have been defined in [5], [12], where main selection of the architectural solutions and design decisions are motivated. The connectivity services management architecture is described in [13]. Figure 1 shows a simplified picture, of the management and control (M&C) plane. The network contains several Core Network Domains (CND), belonging to NPs, and also access networks (AN). The ANs are out of scope of VCANs, given the large variety of technologies and degree of resource management. The CAN layer M&C is partially distributed: one *CAN Manager* (CAN_Mgr) belonging to CANP exists for each IP domain, doing VCAN planning, provisioning, advertisement, offering, negotiation, request for installation and exploitation. Each network domain has an *Intra-domain Network Resource Manager* (Intra_NRM), as the ultimate authority configuring the network nodes. This architectural solution allows an incremental deployment for Network Providers: an NP can be enhanced in order to become also a CAN Provider. The End User (EU) terminals are connected to the network through Home Boxes (HB). The novel CAN routers (not shown here) are called *Media-Aware Network Elements (MANE)* to emphasize their main additional capabilities: content and context – awareness. The CAN layer cooperates with HB and Service Environment by offering them CAN services.

The architectural solution for VCAN Management has been already defined in [13]. Here only a short summary is recalled for sake of clarity. A functional block at SP performs all actions needed for VCAN support (planning, negotiation with CANP, VCAN exploitation) while the CAN Manager at CANP performs VCAN provisioning and operation. The two entities interact based on the SLA/SLS contract initiated by the SP (SLA1 – in the Figure 1). Then the initiator CAN manager (e.g., CAN Manager 1) has to discuss with other involved CAN Managers in hub style (the topology discovery and determination of the other domains involved in a multi-domain VCAN are not subjects of this paper) negotiating with them SLAs (e.g., SLA2.1, SLA 2.2).

Each CAN Manager at its turn has to ask resources from its associated Intra_NRM and establishes with it a SLAs (SLA3.1, SLA3.2 and SLA3.3). To do this, CAN Manager runs a *combined routing, reservation and VCAN mapping algorithm* described in [15]. If the set of SLAs is successful, then the Intra_NRMs receive commands from their CAN Managers to install the configurations in routers, as to assure the QoS characteristics of the VCANs. These actions can be done immediately after SLA agreements or later (conforming to the options expressed in SLA1). Later, the media flow will be transported through the QoS enabled VCAN pipes from content servers to end users. The network technologies supporting these are Multiprotocol Label Switching (MPLS) and/or differentiated services (DiffServ, [5]).

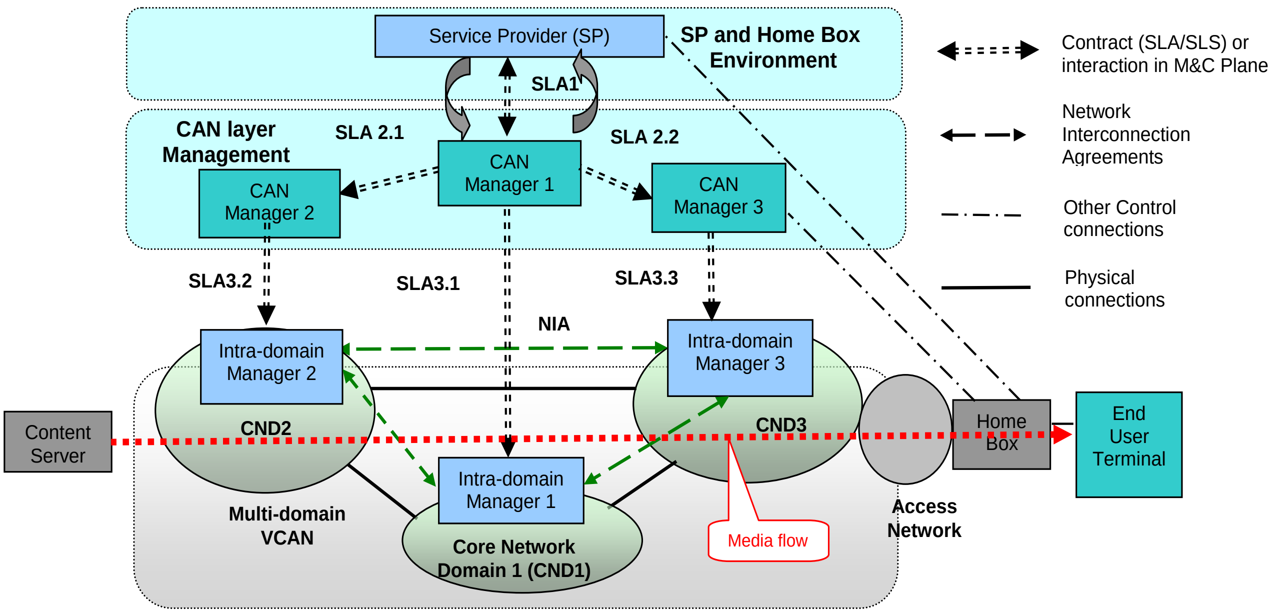

Figure 1. Business Actors and SLA framework high level view

IV. SLA/SLS FRAMEWORK FOR RESOURCE PROVISIONING

The general ALICANTE template of an SLS between the SP and CANP has been defined, able to support more or less sophisticated SP requirements (in practice only a sub-set of the parameters might be specified by the Service Provider).

Table 1 shortly describes the sections of an SLS between an SP and the initiator CAN Manager. Unspecified SLS values (by the SP) let CAN Manager and/or Intra_NRM to apply their own policies. The SLS is transported through negotiation protocols (vertically between SP-CAN Manager and horizontally between initiator CAN Manger and other CAN Managers of the involved core network domains.

TABLE I. TYPICAL SLS (SP-CANP) TEMPLATE EXAMPLE

SLS sections	SLS Element/Clause
General	1. SLS Identification 2. VCAN associated CATI (Content Aware Transport Information) optionally inserted in the data packets by the content servers describes Service Type, Service sub-type, etc.
VCAN Connectivity services requirements needed by SP for VCAN	1. Topology (pipe, hose, funnel) and scope (ingress, egress points). Identifies topology and the edge points of the topological region over which the QoS applies. 2. Connectivity class quantitative/qualitative: Bandwidth, Delay, Jitter, Loss, Availability. 3. VCAN time life, optionally define the time-life of this SLS.
VCAN Traffic Processing Requirements	1. Traffic identification: Ingress flow Id, Egress flow Id, Ingress point, Egress point. 2. QoS guarantees conformance algorithm: Traffic Control (TC) algorithm and parameters. 3. DiffServ-like treatment of excess traffic: dropping, re-marking, shaping, adapting. 4. Routing and forwarding rules: constraints on the way to compute the paths for forwarding. 5. Adaptation requirements: describe the condition (thresholds, etc.), under which the media flow adaptation is allowed for Partially Managed and Unmanaged ALICANTE services.
VCAN Services	1. Monitoring methods: specifies how the SLS should be verified at CAN layer.

Assessment Requirements	2. Monitoring tasks: describes when and how the monitoring actions are performed. 3. Details of notification and reports: time, level of information aggregation, etc.
VCAN Allowed actions	1. Availability and VCAN service schedule: time intervals allowed for service invocation. 2. Invocation methods: conditions of invocation (see also the time life of a VCAN). 3. Modification permission of Connectivity services, Traffic Processing and Service Assessment: describe the modifications allowed for the three categories.

V. DEVELOPMENT AND IMPLEMENTATION RESULTS

This section will present the main implementation decisions for SLA framework and samples of results.

A. *SLS signaling*

Several solutions and scenarios have been analyzed for different cases of VCAN realization. Two step approach has been selected for VCAN construction: *negotiation and implicit resource reservation; VCAN installation in the network routers.* Usually the Service Provider initiates the new VCAN construction. After negotiation and logical resources reservation the SP requests installation (invokes) this VCAN, i.e. asks to CAN Manager (and this asks to Intra_NRM) to install the VCAN appropriate configurations in the routers. A single SLS per VCAN, or several ones, can be negotiated if these SLSs define the same QoS and security class.

One basic signaling case (described below) is for a unicast communication VCAN, where the virtual VCAN pipes are implemented on top of MPLS+Diffserv enabled paths. To decide the final implementation solution several signaling variants have been analyzed and compared.

a. Information on network resources (controlled by Intra-NRM) are obtained by the CAN Manager on demand, i.e. asynchronously, when a new SP request arrives. The advantages are: network availability updates in the CAN_Mgr Data Base are done only when needed, i.e. at each new SP request (for initiator CAN_Mgr), or at request of other CAN_Mgrs. The Intra_NRM does not care to inform CAN_Mgr each time when it makes its *Resource Availability Matrix (RAM)* modifications. However, a higher signaling overhead exists: each time that a new request comes to this CAN_Mgr, from SP or other CAN_Mgrs, the RAM information should be asked for each domain.

b. *Network RAM delivered proactively to CAN Manager, asynchronously, by Intra_NRM*.

The RAM is proactively uploaded to CAN_Mgr at Intra_NRM initiative, (when modifications appear, or periodically).. Less signaling is needed when new SP requests (or request from other CAN_Mgrs) arrive at this CAN_Mgr (the RAM information is already available). However, tighter synchronization Intra_NRM – CAN_Mgr, is necessary, given that Intra_NRM might change RAM info while CAN_Mgr is solving some computation based on previous RAM values.

In the current implementation, the solution *on demand* has been selected. Figure 2 presents the signaling actions implemented in case of a multi-domain VCAN requested by the SP to CAN_Mgr1 (VCAN initiator). It has been supposed that CAN_Mgr2 and 3 should participate to the VCAN construction, spanning three core network domains. Similar diagrams have been designed and verified in implementation, e.g. for multicast VCANs, [15].

c. An advanced solution is necessary when different VCANs have different life-time; the resource reservation should consider such intervals. To simplify the management, a time-unit T can be defined and a VCAN time-life can be kT, with k natural number. The VCAN construction and/or reconfiguration happens at nT instants only (the T value can be selected by the Can Manager policy - e.g., few minutes). At each nT instant one can update the RAM information in CAN Manager, or asynchronously (at Intra_NRM initiative) when major changes happens to its RAM.

The implementation technology used in signaling framework is shortly described below. The CAN Manager has been implemented by using *Python* language. This allows to deploy the applications on multiple Operating Systems and to have a self contained system that can be installed without affecting the rest of the applications installed. *Virtual environments* technology that comes with Python has been used. For database access we use an ORM (Object Relational Mapper), *SqlAlchemy*, which helps to hide the details while having the full power to interact with the data.

The CAN Manager interaction to other entities (SP, Intra_NRM, or other CAN Managers) is done using web services. Several frameworks for SOAP have been analyzed; *Suds* has been selected for initiating web service requests and *Spyne* for listening to requests. The application is presented as a *Web Server Gateway Interface* (WSGI) object, a Python protocol standard for web applications, and can be deployed on any WSGI capable container. We are mainly using *Spawning* as a container started as a service at system startup, but we also tested on the *nginx web server* with *uWSGI* as container. The OS we deployed this on is a Debian distribution of GNU/Linux.

The CAN Manager listens for SP requests and then continues by running its internal algorithms. After computing a VCAN mapping for the input RAM, it sends the requests to all other CAN Managers involved in the VCAN. In the sense of SOAP specification the CAN Manager has two listening ports: one for SP requests and another one for other CAN Managers requests. The CAN Manager may issue requests to either of two entities: the Intra_NRM, and other CAN Managers. The database is accessed using database specific protocols.

A possible implementation improvement is to make asynchronous all calls between entities. However this involves a lot of glue multi-threading code; so for the current implementation synchronous calls have been selected.

 ISBN: 978-1-61208-279-0

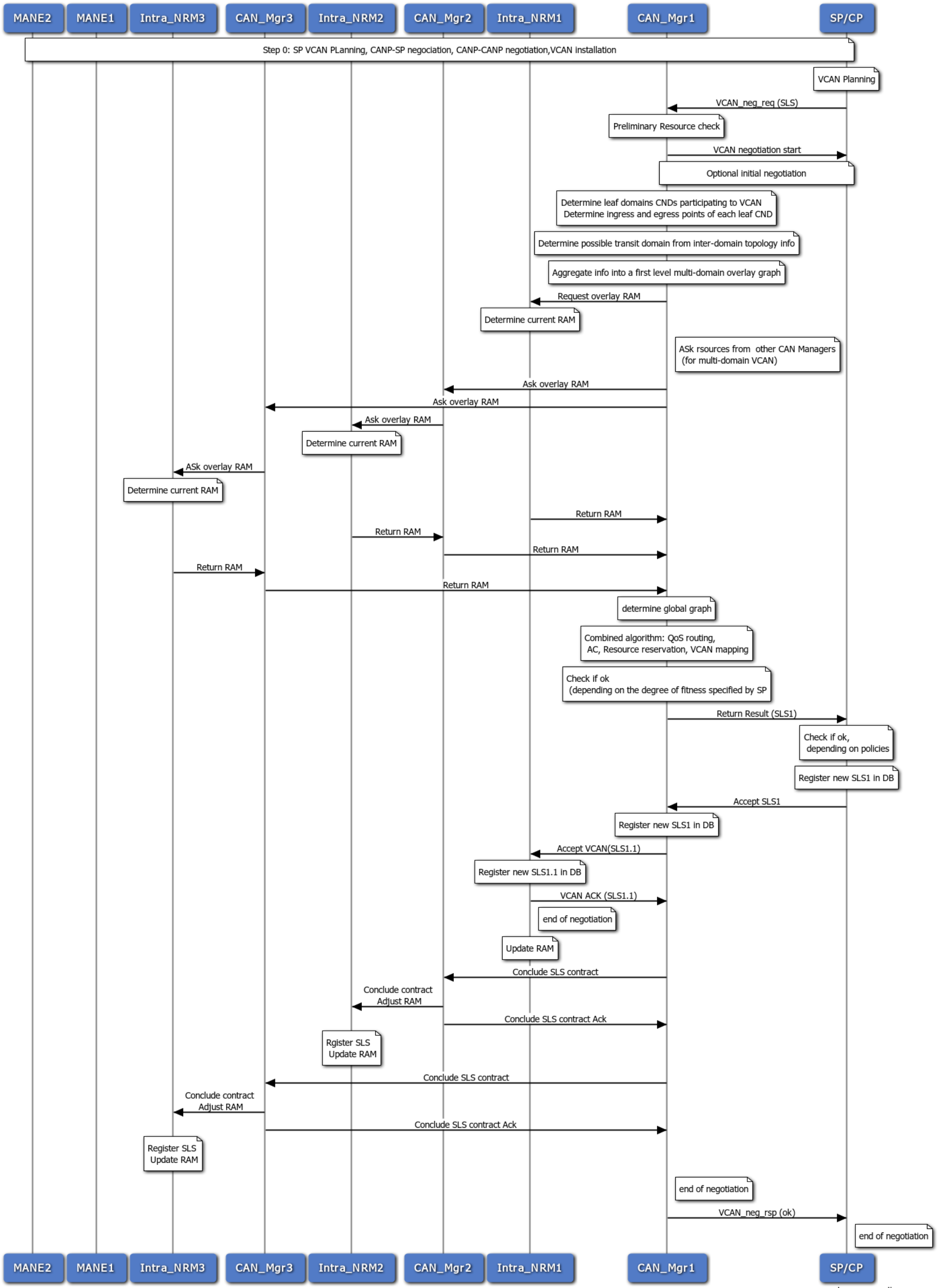

Figure 2. Implementation solution of the signaling for multi-domain unicast VCAN (MPLS based)

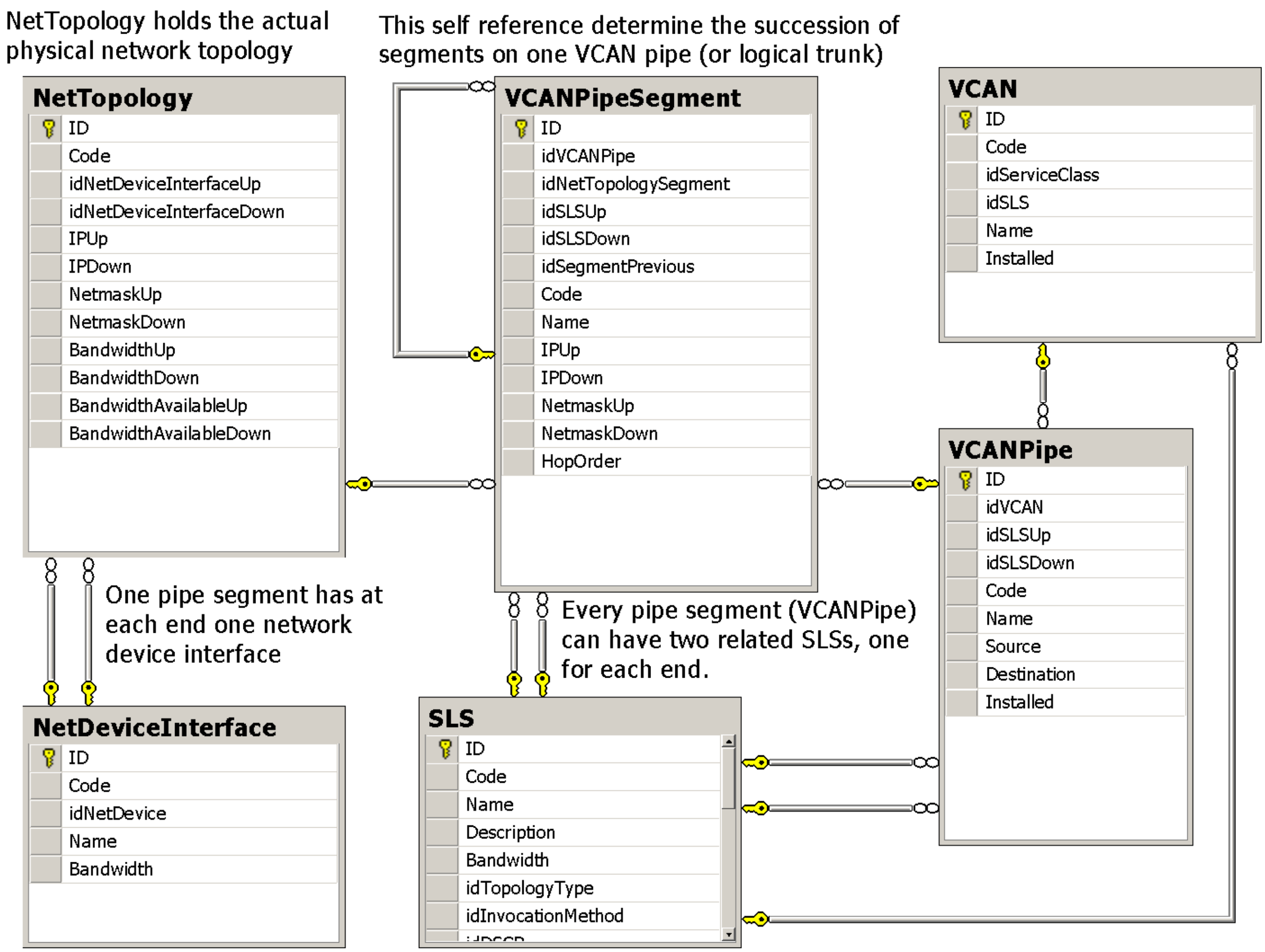

Figure 3. Sample of the Data Base at CAN Manager: Table *VCANPipe* and its connections with neighbor tables

B. *Data Base at CAN Manager*

A relational database has been designed and implemented around several main building blocks or tables: *"SLS"* for SLS templates used to hold negotiated resources around the network, *"NetTopology"* holds the image of actual physical network topology on which VCAN pipelines are built, *"VCANPipeSegment"* for VCAN Pipe Segments and *"NetDeviceInterface"* for information about network path endpoints known by the entity using this database (SP or CAN_Mgr). It is not mandatory for an endpoint to represent the actual physical interface located on network devices.

Figure 3 presents an overview of the current Database implementation that holds only basic SLS parameters (like *Bandwidth*, *Delay*, *Loss* or *Jitter*) in the main SLS table, other parameters suitable in some category will be hosted by satellite tables having prefix "SLS_". There are a couple of reasons for this design: it is much easier to further expand the list of SLS parameters compared to the other design where everything is stored in one big table, better readability of the data structures for someone who knows the design documentation, hierarchical organization.

CAN Manager database stores and provides access to an overlay image of the provisioned MPLS pipelines. This is translated into a matrix of the managed network domains by Intra_NRM having vector elements of type *{..., (input_router_addr, output_router_addr, Bandwidth, x, y), ...}* where parameters x and y are intended for future usage. Every vector element is the concrete representation of one MPLS Label Switched Path (LSP) with a minimum of information as terminal end points and assigned bandwidth.

Data Base contains four different table types according to their role in the architecture; therefore the stored information is specific to some: *Entity (SLS, VCAN, VCANPipe, NetDevice, NetDeviceInterface)*, satellite information related to some entity (*SLS_QoS, SLS_ServiceClass, SLS_CATI*), classification items that enumerate the available entity types (*NetDeviceType, TopologyType*), relationship between entities (*VCANxSLS, SLSxCATI*). The Data Base at CAN_Manager has an interface with the external world (i.e. other functional modules of the CAN_Manager or even

 ISBN: 978-1-61208-279-0

external); for this reason an embedded application programming interface (API) has been implemented that provides CRUD (*create, read, update and delete*) functionality, data validation, automation procedures, path computations, activity logging and more. In terms of security, the Data Base design provides some advantages in hiding important information about VCAN network to the outside world.

All active modules (including the VCAN – related algorithms, i.e. routing, resource reservation and mapping) and Data Base have been implemented. The message sequence chart presented in Figure 2 and other similar scenarios (e.g. for multicast VCANs) have been validated.

VI. CONCLUSIONS AND PERSPECTIVES

This paper presented an SLA/SLS framework development for dynamic Virtual Content Aware Network (VCAN) provisioning, spanning multiple core network domains. The SLAs are customized as to satisfy Service Providers requirements for media distribution services.

The architectural and implementation solutions presented have several advantages. A light virtualization solution for multi-domain media distribution QoS enabled is offered, creating a cross-layer optimization loop between the transport and application layers. The Intra-domain network resource manager independence (important business requirement) is preserved, due to two-level hierarchical architecture (CAN and network layers). All tasks to construct multiple domain VCANs are delegated to CAN Provider (Service Providers are just asking for VCANs and then using them). Seamless deployment is possible: VCAN guaranteed services can coexist with best effort ones (families of SLS templates can be defined). The CAN Manager software can be naturally installed as an upgrade of the Intra_NRM, thus transforming an NP in a CAN Provider.

The management system is scalable given that the VCANs are installed as “networks”, on demand, on short-mid-long term, and not at per-flow demand; therefore the signaling tasks do not create a significant overhead.

The architecture is similar to the new approach of Software Defined Networking (SDN) [16], [17] where important control functions are moved out of data plane into a separate control plane. This decoupling enables both planes to evolve independently, and brings advantages such as high flexibility, vendor-agnosticism, programmability, and the possibility of realizing a centralized network view.

The implementation of the proposed system is in final phase in the ALICANTE project. Future work may envisage the study of a migration towards a solution closer to SDN.

Acknowledgments

This work was supported partially by the EC in the context of the ALICANTE project (FP7-ICT-248652) and partially by the projects POSDRU/107/1.5/S/76909.

REFERENCES

[1] J. Schönwälder, M. Fouquet, G .Dreo Rodosek, and I.C Hochstatter, “Future Internet = Content + Services + Management”, IEEE Communications Magazine, vol. 47, no. 7, Jul. 2009, pp. 27-33.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet Impasse through Virtualization”, Computer, vol. 38, no. 4, Apr. 2005, pp. 34–41.

[3] 4WARD, “A clean-slate approach for Future Internet”, http://www.4ward-project.eu/ (last access May 2012).

[4] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Communications Magazine, March 2011.

[5] ALICANTE, Deliverable D2.1, ALICANTE Overall System and Components Definition and Specifications, http://www.ict-alicante.eu, Sept. 2011.

[6] M. P. Howarth, P. Flegkas, G.Pavlou, N.Wang, P.Trimintzios, D.Griffin, J.Griem, M.Boucadair P.Morand,A.Asgari, P.Georgatsos, "Provisioning for Interdomain Quality of Service: the MESCAL Approach", IEEE Communications Magazine, June 2005, pp.129-137.

[7] P. Levis, M. Boucadair, P. Morrand, and P. Trimitzios, “The Meta-QoS-Class Concept: a Step Towards Global QoS Interdomain Services”, Proc. of IEEE SoftCOM, Oct. 2004.

[8] ENTHRONE, End-to-End QoS through Integrated Management of Content, Networks and Terminals, FP6 project, www.ist-enthrone.org/ (last access May 2012).

[9] M. Boucadair, P. Lévis,D. Griffin, N. Wang, M. Howarth, G. Pavlou, E. Mykoniati,P. Georgatsos, B. Quoitin, J. Rodríguez Sánchez, M. L. García-Osma “A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets”, IEEE Communications Magazine, Sept. 2007, pp. 134-143.

[10] F.J. Ramón Salguero, “COntent Mediator architecture for content-aware nETworks”, 2010, http://www.comet-project.org/.

[11] W.K. Chai, et.al., CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services , IEEE Communications Magazine, March 2011.

[12] H. Koumaras , D. Negru , E. Borcoci , V. Koumaras , C. Troulos , Yes. Lapid , E. Pallis , M. Sidibé ,G. Gardikis, G. Xilouris, C. Timmerer, „Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks” FIA Book 2010

[13] E. Borcoci, M. Stanciu, D. Niculescu, D.Negru, G. Xilouris, “Connectivity Services Management in Multi-domain Content-Aware Networks for Multimedia Applications ” , Proc. of. INTERNET 2011, Luxembourg, June 2011, www.iaria.org.

[14] E.Borcoci, G.Carneiro, R. Iorga**,** “Hybrid Multicast Management in a Content Aware Multidomain Network**”,** AFIN 2011, The Third International Conference on Advances in Future Internet, http://www.thinkmind.org/index.php?view=article&articleid=afin_2011_5_20_70107

[15] E.Borcoci, R.Miruta, S.Obreja, “Multi-domain Virtual Content-Aware Networks Mapping on Network Resources”, EUSIPCO Conference, 27-31 August 2012, Bucharest, http://www.eusipco2012.org/home.php

[16] S.H.Yeganeh, A.Tootoonchian, Y.Ganjali, “On Scalability of Software-Defined Networking”, IEEE Communications Magazine, February 2013, pp.136-141.

[17] B.Raghavan, T.Koponen, A.Ghodsi, M.Casado, S. Ratnasamy, S.Shenker, “Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure”, HotNets-XI Proceedings of the 11th ACM Workshop on Hot Topics in Networks, 2012, pp. 43-48, doi: 10.1145/2390231.2390239.