Content-aware networking in virtualised environments for optimised resource exploitation

Petros Anapliotis

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHEMATIQUES et INFORMATIQUE

SPÉCIALITÉ INFORMATIQUE

Par Petros ANAPLIOTIS

Content-Aware Networking in Virtualised Environments for Optimised Resource Exploitation

Approche Réseau basée sur la Conscience du Contenu pour l'Optimisation de l'Exploitation des Ressources au sein d'Environnements Virtualisés

Sous la direction de : Daniel NEGRU
co-directeur : Evangelos PALLIS

Soutenue le 19/12/2014

Membres du jury :

Mr Xavier Blanc	Professeur, Université de Bordeaux, France	Président du jury
Mr Harrry Perros	Professeur, Université de North Carolina, USA	Rapporteur
Mr Adlen Ksentini	Maître de Conférences HDR, Université de Rennes, France	Rapporteur
Mr Anastasios Kourtis	Directeur de Recherche, NCSR Demokritos, Grèce	Examinateur
Mrs Paraskevi Fragopoulou	Professeur, Technological Educational Institute of Crete, Grèce	Examinateur
Mr Daniel Negru	Maître de Conférences HDR, Université de Bordeaux, France	Directeur de thèse
Mr Evangelos Pallis	Maître de Conférences, Technological Educational Institute of Crete, Grèce	Co-directeur de thèse

Titre : Approche Réseau basée sur la Conscience du Contenu pour l'Optimisation de l'Exploitation des Ressources au sein d'Environnements Virtualisés

Résumé : Aujourd'hui, l'hétérogénéité des infrastructures de réseaux actuelles, ainsi que le manque d'interopérabilité en termes d'architectures et de cadres pour l'adaptation du contenu aux contextes des différents utilisateurs, empêchent les Prosumers (consommateurs-fournisseurs) d'offrir une haute qualité d'expérience sur différentes plates-formes et au travers de contextes diversifiés. Par conséquent, l'objectif de cette thèse est d'étudier, concevoir et développer une architecture novatrice, susceptible d'offrir la QoS/QoE garantie en exploitant efficacement les ressources disponibles et en adaptant dynamiquement la performance du réseau selon les environnements Réseau, Service et Utilisateur. Pour cela, l'architecture proposée est basée sur (1) un cadre de gestion distribuée qui exploite des mécanismes de réseau conscient du contenu pour identifier le contenu en transit et la correspondance sur les exigences de QoS/QoE, et sur (2) un mécanisme d'allocation des ressources de réseau et leur adaptation aux caractéristiques de QoS/QoE demandées. Un prototype de routeur de contenu a été réalisé, offrant des fonctions de reconnaissance du type de contenu et de routage suivant le contenu. Il propose un système de gestion synergique capable d'orchestrer les processus d'optimisation cross-layer pour les services de différenciation/classification et à termes une exploitation efficace des ressources. La validité de l'architecture proposée est vérifiée par un grand nombre d'expériences menées à l'aide d'infrastructures physiques et virtuelles. Un banc d'essai à grande échelle conforme aux spécifications de conception architecturale a été déployé pour valider l'approche proposée.

Mots clés : Réseau de contenu, virtualisation, gestion de ressources, QoS

Title : Content-Aware Networking in Virtualised Environments for Optimised Resource Exploitation

Abstract : Today, the heterogeneity of current networking infrastructures, along with the lack of interoperability in terms of architectures and frameworks for adapting content to the various users' contexts, prevent prosumers to deliver high QoE over different platforms and under diversified contexts. Consequently, the objective of this PhD thesis is to study, design, and develop a novel architecture capable to offer guaranteed QoS/QoE by efficiently exploiting the available resources and by dynamically adapting the network performance across the various Service, Network and User environments. To this end, the proposed architecture is based on (1) a distributed management framework that exploits Content Aware Network (CAN) mechanisms – on top of the Internet Protocol (IP) – for identifying content in transit and mapping its QoS/QoE requirements into specific network characteristics, and on (2) a network resource allocation mechanism for adapting the intra-domain resources to the requested QoS/QoE. A prototype Media-Aware Network Element (MANE) has been achieved, offering content type recognition and content-based routing/forwarding as a matter of guaranteed QoS/QoE provision in an end-to-end approach. Furthermore, it proposes a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification, towards efficient resource exploitation. The validity of the proposed architecture is verified through a large number of experiments conducted using physical and virtual infrastructures. A large-scale test-bed conforming to the architectural design specifications was deployed for validating the proposed approach.

Keywords : Content-Aware Networks, virtualisation, resources management, QoS

Unité de recherche

Laboratoire LaBRI, UMR 5800

351, Cours de la Libération 33405 Talence Cedex, France

This Ph.D. Thesis is dedicated to my beloved family

RESUME

Au cours des dernières années, il y eut une croissance explosive dans le développement et le déploiement de réseaux (fixe et mobile) et des technologies multimédia, visant à aider les citoyens à la création, la consommation et le partage de contenus multimédia à partir de tout lieu et à tout moment. Ce nouveau type d'utilisateur, à la base consommateur traditionnel mais maintenant également devenu fournisseur implicite, constitue de nos jours une nouvelle forme d'acteur média, communément appelé «Prosumer ». En utilisant des équipements plus sophistiqués mais abordables, les Prosumers peuvent générer et manipuler le contenu de toutes qualités, de la HD à la 3D interactive et autres images stéréoscopiques. En tant que fournisseurs réels de contenus, les Prosumers doivent également être en mesure de fournir des services de qualité garantie (QoS) pour tout éventuel consommateur, en plus de maintenir la livraison de contenu de manière évolutive, fiable, efficace et interopérable, pour maximiser la qualité de l'expérience (QoE) des utilisateurs.

La diffusion de médias sans coupure sur les infrastructures de réseau existantes soulève un certain nombre de défis. D'une part, les services doivent devenir plus sensibles au contexte pour permettre leur livraison aux environnements d'accès diversifiés. D'autre part, les plates-formes de diffusion sur IP de médias actuelles doivent se mettre au niveau afin de répondre au nombre sans cesse croissant d'utilisateurs, en plus d'être en mesure d'adapter leurs ressources et les performances disponibles pour un maintien de la qualité (QoS / QoE) au niveau maximal. En d'autres termes, l'hétérogénéité des infrastructures de réseau actuelles d'une part, ainsi que le manque d'architectures et de cadres pour l'adaptation du contenu aux contextes des différents utilisateurs réseaux d'autre part, empêchent aujourd'hui les Prosumers d'offrir une haute qualité d'expérience sur différentes plates-formes et au travers de contextes diversifiés.

A partir de ce constat, l'objectif principal de cette thèse est d'étudier, concevoir et développer une architecture novatrice, susceptible d'offrir une Qualité de Service (QoS) et d'Expérience (QoE) garantis, en exploitant efficacement les ressources disponibles et en adaptant dynamiquement la performance du réseau au travers des environnements Utilisateurs, de Réseaux et de Service. Pour y parvenir, l'architecture proposée est basée sur (1) un cadre de gestion distribuée qui exploite les mécanismes réseaux conscients du contenu (Content-Aware Networks - CAN) - au-dessus du protocole Internet (IP) - pour identifier le contenu en transit et établir la correspondance par rapport aux exigences de QoS/QoE, et (2) un mécanisme d'allocation des ressources de réseau pour adapter les ressources intra-domaines aux caractéristiques de QoS/QoE demandés.

S'appuyant sur les progrès récents dans les techniques d'identification du trafic réseau et en élaborant des mécanismes de signalisation de contenu explicites, la thèse présente un prototype d'un routeur de bordure basé sur la conscience du contenu appelé Media-Aware

Network Element (MANE). Ce dernier propose des fonctions (1) de reconnaissance du type de contenu et (2) de routage/transfert en fonction du contenu, le tout à des fins de garanties de provision de QoS/QoE dans une approche de bout-en-bout. En outre, il propose un système de gestion synergique capable d'orchestrer les processus d'optimisation cross-layer pour les services de différenciation/classification et à termes une exploitation efficace des ressources. Afin de vérifier la validité de l'architecture proposée, un certain nombre d'expériences ont été conçues et mises en œuvre en utilisant des infrastructures physiques et virtuelles. Un banc d'essai conforme aux spécifications de conception architecturale a été mis en œuvre et a également servi en tant qu'environnement expérimental pour réaliser des mesures d'évaluation de la performance et finalement valider l'approche proposée.

ABSTRACT

Over the recent years, there has been an explosive growth in the development and deployment of network (fixed and mobile) and multimedia technologies, aiming to assist citizens in the creation, consumption and sharing of audiovisual content from any place and at any time. This new kind of user, stemming from traditional consumer but being also able to act as implicit provider, constitute nowadays a new form of media players, commonly known as "Prosumers". By utilising sophisticated but affordable end-user equipment, prosumers can generate and manipulate high quality content ranging from high definition video to interactive 3D and stereoscopic images. Towards becoming actual providers, prosumers must also be able to deliver services with guaranteed quality (QoS) to any potential consumer, besides maintaining content delivery in a scalable, reliable, efficient and interoperable way, for maximising users' Quality of Experience (QoE).

To this extent, seamless media delivery over existing networking infrastructures raises a certain number of challenges. On one hand, services have to become more context-aware to enable their delivery to large and diversified access environments. On another hand, current Internet media delivery platforms have to scale up in order to meet the continuously increasing number of users, besides being able to adapt their available resources and performance towards keeping quality (QoS/QoE) at maximum level. In other words, the heterogeneity of current networking infrastructures, along with the lack of interoperable network architectures and frameworks for adapting content to the various users' contexts, prevent prosumers to deliver high QoE over different platforms and under diversified contexts.

In this direction, the main objective of this PhD thesis is to study, design, and develop a novel architecture capable to offer guaranteed QoS/QoE by efficiently exploiting the available resources and by dynamically adapting the network performance across the various Service, Network and User environments. To this end, the proposed architecture is based on (1) a distributed management framework that exploits Content Aware Network (CAN) mechanisms – on top of the Internet Protocol (IP) – for identifying content in transit and mapping its QoS/QoE requirements into specific network characteristics, and on (2) a network resource allocation mechanism for adapting the intra-domain resources to the requested QoS/QoE.

Building upon the recent advances in traffic identification techniques and by elaborating on explicit content signalling mechanisms, the thesis presents a prototype Media-Aware Network Element (MANE) (i.e. an enhanced content-aware network module) that offers content type recognition and content-based routing/forwarding as a matter of guaranteed QoS/QoE provision in an end-to-end approach. Furthermore, it proposes a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification, towards efficient resource exploitation. In order to verify the validity of the proposed architecture, a number of experiments were designed and

conducted using physical and virtual infrastructures. For this reason, a test-bed conforming to the architectural design specifications was implemented, which also served as experimental environment for carrying out performance evaluation measurements and ultimately validating the proposed approach.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Dr. Daniel Negru, Associate Professor in the LaBRI lab., CNRS, University of Bordeaux, for his constant guidance, his valuable comments and prompt suggestions, as well as for his continuous mentorship towards the successful completion of this Ph.D. research.

I also would like to express my appreciation to my research advisor Dr. Evangelos Pallis, Associate Professor in the Department of Informatics Engineering at Technological Educational Institute of Crete, for his constant support and thoughtful guidance, his targeted observations and thorough comments, as well as for his continuous supervision during these years of my Ph.D. research.

My special thanks to my closest research colleagues and friends, Mr. Evangelos Markakis and Mr. Anargyros Sideris, for their support and valuable assistance during the progress of this research.

I would like to thank my parents and my brothers, who encouraged and helped me at every stage of my personal and academic life.

Finally, I would like to express my sincere thanks to my wife Evi Karataraki for her love, understanding and endless patience.

.

TABLE OF CONTENTS

RESUME..V

ABSTRACT...VII

ACKNOWLEDGMENTS...IX

TABLE OF CONTENTS...X

LIST OF FIGURES...XIII

LIST OF TABLES..15

1. INTRODUCTION..16

1.1. Background and statement of the problem 16

1.2. Aim of the research and contribution.. 17

1.3. Structure of the thesis 18

2. STATE OF THE ART REVIEW AND RELATED WORK...20

2.1. Introduction 20

2.2. Current state in the Network Media Content Value Chain 21

2.2.1. Role of Content Creators 21

2.2.2. Role of Content Providers 22

2.2.3. Role of Service Providers.. 22

2.2.4. Role of Network Providers 24

2.2.5. Role of End Users 25

2.3. Current advances in the Content Delivery Architectures 25

2.3.1. Content Delivery Networks (CDN) 25

2.3.2. P2P Networks.. 26

2.3.2.1. Centralized Systems 26

2.3.2.2. Decentralized Systems 27

2.4. Current advances in Virtualized Environments 27

2.4.1. Network Virtualization 27
2.4.2. Software Defined Networking (SDN) 28
2.4.2.1. SDN against Current Network Limitations 28
2.4.2.2. SDN architecture approach 29
2.4.2.3. Software Defined Internet Architecture 30
2.5. Summary 31
3. AN ARCHITECTURAL PARADIGM TOWARDS FUTURE INTERNET 32
3.1. Introduction 32
3.2. The ALICANTE approach 32
3.2.1. The high-level architecture overview 32
3.2.2. Overall system architecture 34
3.2.3. ALICANTE architecture and Future Internet standardization efforts 35
3.2.4. ALICANTE compared to other existing Future Internet architectures 37
3.3. Summary 40
4. MANE: A CONTENT-AWARENESS ENABLER IN FUTURE MEDIA NETWORKS 41
4.1. Introduction 41
4.2. Deep Packet Inspection (DPI) 42
4.3. Content-Aware Transport Signalling 44
4.4. Content-Aware Network Elements in ALICANTE 49
4.4.1. Flow-Awareness 49
4.4.2. MANE General Architecture 51
4.4.3. Functional Blocks and Interfaces 52
4.5. MANE Evaluation 54
4.5.1. Scalability of MANE operations 59
4.6. Summary 60
5. A NETWORK RESOURCE MANAGEMENT FRAMEWORK IN VIRTUAL CONTENT-AWARE INFRASTRUCTURES 61
5.1. Introduction 61
5.2. The CAN & Network Subsystem 62
5.2.1. The Intra-domain Network Resource Manager Architecture 64

5.2.1.1. Network Planning and Provisioning 65

5.2.1.2. Intra-domain Traffic Engineering and Resource Provisioning............... 69

5.2.1.3. Network Layer Monitoring 73

5.2.2. Validation Environment 74

5.2.2.1. Validation Results 76

5.3. Summary.. 84

6. CONCLUSIONS 85

6.1. Overview 85

6.2. Thesis contribution to knowledge and research.. 85

6.3. Fields for future exploitation.. 86

APPENDIX A..88

List of acronyms 88

APPENDIX B..91

Publications 91

C1. Published papers in international conferences and workshops................................ 91

C2. Reports.. 91

REFERENCES 93

LIST OF FIGURES

Figure 1 The current Networked Media Content Value Chain - simplified representation 21

Figure 2 ALICANTE High level Architecture Overview 33

Figure 3 High-level functional architecture of the ALICANTE system.......................................35

Figure 4 FMIA-TT Reference Model..36

Figure 5. RTP Header Syntax 47

Figure 6. RTP Header Extension Syntax – 2B 47

Figure 7. RTP Header Extension Syntax – 1B 48

Figure 9. Packet path through the MPLS ingress LER 52

Figure 10. MANE architecture description..54

Figure 11. MANE related Experiment Setup 55

Figure 12. Packet Forwarding Performance of MANE..55

Figure 13. MANE Capturing Performance...57

Figure 14. MANE Forwarding Performance...58

Figure 15. Aggregated CPU Load 58

Figure 16. Packet capturing performance versus new flow density..60

Figure 17. CAN and Network layers in ALICANTE...63

Figure 18. IntraNRM Functional Architecture...65

Figure 19. MPLS QoS-based Label distribution..70

Figure 20. Monitoring at CAN and Network layers and associated modules 73

Figure 21. Testing Environment (Alicante Pilot topology in Bordeaux) 76

Figure 22. LSPs Table from netDB...77

Figure 23. Interfaces Table from netDB 78

Figure 24. Web-interface of Intra-NRM/LSP representation 78

Figure 25. Routes Table from netDB..79

Figure 26. LSP installation output with label 1000 79

Figure 27. Output after installing DiffServ with MPLS 81

Figure 28. Traffic trace MANE2 eth3 82

Figure 29. Traffic trace MANE2 eth0 82

Figure 31 Traffic trace in EF class queue LSR2 eth0 83

Figure 32. Traffic trace - label decapsulation SW-MANE1..83

LIST OF TABLES

Table 1 FMIA-TT and ALICANTE Architectures Mapping 37

Table 2 Comparison of ALICANTE with other FI oriented projects 40

Table 3. CATI header structure 44

Table 4. Content-Aware Transport Information (CATI) and semantics. 44

Table 5. Summary of characteristics of CATI insertion methods 46

Table 6. Flow Hash Table of the software-based flow-awareness module 51

Table 7. Throughput of MANE and IP Router 56

Table 8. Multiple traffic class to single LSP mapping 71

Table 9. Single QoS class per LSP. 71

Table 10. Classes of Services 71

Table 11. AF Classes of services 72

Table 12. DSCP to EXP mapping 81

1. INTRODUCTION

1.1. Background and statement of the problem

Recent advances in multimedia technologies, as well as the continuous technological enhancements of end-user equipment in order to support them, have led to the emergence of a bundle of services that are intrinsically characterized by the user participation both in content-generation and in manipulation processes. By exploiting sophisticated high performance, but affordable, audio/video capturing, editing and representation modules, citizens can nowadays develop high quality content, ranging from standard resolution video to high definition and/or 3D images. At the same time, the wide-spread adoption of social media platforms such as Facebook, YouTube, Flickr, Instagram etc., enabled citizens to share, distribute or even merchandise their high quality media, thus gradually transforming them from passive information-consumers to implicit service/content providers (i.e. Prosumers). Establishing, however, prosumers to fully operational providers requires a networking infrastructure capable not only to host high quality content/services, but also and most predominant provide the same Quality of Service (QoS) and Quality of Experience (QoE) at the End-Users' premises; an issue that current network technologies can hardly confront.

Since current infrastructures are mainly based on the Internet best-effort model, they cannot satisfy the strict QoS/QoE requirements that contemporary services pose. Evidently, the current Internet architecture and the corresponding technologies are nowadays posing significant challenges with regard to the provision of multimedia services, especially when QoS/QoE comes to the foreground. Taking into account that Internet was initially designed as a content/service agnostic infrastructure for resilient and opportunistic delivery of non-real-time data, quality was a matter only of the network/transport layers, which cannot inherently support adaptation and customization of the available resources towards meeting the content/services' requirements. Towards this direction, several cross-layer techniques have been proposed, enabling the various layers to interact with each-other, adapt their resources and customize their performance according to the requested QoS/QoE. However, most of these techniques are technology dependent "add-on" solutions to the existing Internet architecture and restricted in "fenced" managed networks with centralized resource management (i.e. intra-domain), thus raising systems' complexity, besides becoming redundant in heterogeneous networking environments. Therefore, the current debates are focused on revisiting design principles and establishing a novel Internet architecture (the so-called Future Internet – FI), able to accommodate inter-domain services provision with guaranteed QoS/QoE provision in an end-to-end approach and over heterogeneous infrastructures.

1.2. Aim of the research and contribution

In the abovementioned challenging and profoundly evolving context, the aim of this PhD thesis is the proposal of a network management system where both content awareness and network virtualization concepts are inherently supported within a Future Internet capable architecture. The challenge is to find a synergetic system architecture and the corresponding mechanisms that can optimally allocate the available network (physical or virtual) resources to diversified multimedia applications, along with the appropriate network-layer interfaces. The ultimate objective is to enable the underlying network to dynamically adapt its behavior according to changing network conditions.

Considering the aforementioned challenges, this thesis introduces a novel architecture, aiming to facilitate a collaborative framework for sharing and consuming Media Services within the Future Internet (FI), constituting an integrated environment where End-Users and Service Providers share and distribute state of the art multimedia-based services. The proposed architecture relies on two main directions: (a) an innovative Service Environment (SE), involving an overlay of interconnected media-centric Home Gateways ("Home-Boxes") and (b) an enhanced Network Environment (NE), featuring inherent Content Awareness through the creation of a virtual overlay network for media transport. On top of the SE, a flexible User Environment (UE) is foreseen to enable ubiquitous service access in various usage scenarios using different wired and wireless terminals, a context modeling and management solution and real-time Quality-of-Experience (QoE) monitoring.

In this thesis, the focus is put on the network side and more specifically on this new overlay layer where Content-Awareness is enabled, the CAN (Content-Aware Network) layer. The research objectives are:
(a) To study identification methods and their respective levels of accuracy for a variety of flows, enhance them using explicit signaling mechanisms (i.e., Content Aware Transport Information – CATI) and integrate them into a Media Aware Network Element (i.e., an enhanced edge router with content awareness capabilities – MANE)
(b) To elaborate a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification towards efficient resource exploitation.

The content awareness feature is considered as a major innovative asset for the next generation of media delivery platforms. There are several terms that can be found in the literature and in associated works in a non-uniform way and have overlapping semantics to indicate a different "networking" behavior: Content-Aware Networking (CAN), Content Oriented Networking (CON), Content-Centric Networking (CCN), Service Oriented Networking (SON) [1][2]. More specifically:

1. CAN can be considered as a very general characteristic meaning that the overlay network may have one basic content awareness feature or a more elaborated one, as below:

    a. ***Content-type awareness:*** it performs (in the network elements) appropriate processing for flows having different content-types (e.g. assuring QoS in multiple domain planes, forwarding, adaptation, security, etc.), based on the content-type learned from the packets themselves and/or in combination with the Management and Control plane information. For example, a set of

VoD flows is considered as having VoD-type and only quantitative differences might exist between individual flow requirements.

b. ***Content-objects awareness:*** named CON in [3][4], it has a finer granularity; it assumes that content-objects are individually recognized and processed accordingly (e.g. name-based routing, content-object caching, etc.)[5][6]. It basically deals with *content objects* upon which it applies the following operations: *naming, locating/routing, delivery/dissemination, and in-network caching.* Here, the routing is based not on "location" as in classic IP but on "name". The CCN approach [7], is a particular case of CON.

This first approach has less granularity, but has the advantage of permitting an incremental evolutionary architecture and implies less additional overhead in the routers. A straightforward mapping existing technologies (eg MPLS/DiffServ) is possible. The second approach actually includes the first. It has a finer granularity, and it can be seen as a "more revolutionary" one. However, it involves much more processing tasks in the network nodes (and significant additional memory if caching is used). So, media object awareness in the routers requires much more resources and creates scalability problems not yet clarified. Also the management and control plane is significantly different, (w.r.t. the first approach) not only at network layer but also at the service layer.

2. SON is an emerging architecture that enables network devices to operate at the application layer with features such as offloading, protocol integration, and content-based routing. By adding application-awareness into the network fabric, SON can provide vast gains in overall performance and efficiency and enable the integration of heterogeneous environments. Among others, SON functionality provides three key benefits: service virtualization, locality exploitation and improved manageability [8]. In this sense, SON is overlapping with CAN and CON.

In this thesis, the architectural approach chosen is inspired from Option 1: content-type awareness. The proposed solution consists of the introduction of content-related metadata information for transport, called CATI (Content-Aware Transport Information), inside media flows by Service Providers and recognition of these information by the network nodes in order to perform their functions accordingly (based on pre-defined agreements). The whole process is orchestrated by a synergetic management system that takes into account content-related information and configures network modules accordingy.

1.3. Structure of the thesis

Following this introductory chapter, the rest of this thesis is structured as follows.

Chapter 2, "State of the Art Review and Related Work" presents the radical evolution in the Network Media Delivery Chain and the role handled by the main actors, i.e., the Content Creators, the Content Providers, the Service Providers, the the Network Providers and finally the End-Users, acting as Content Consumers but also as Providers. This part is followed by the state of the art of multimedia distribution networks, focusing on unmanaged best effort delivery platforms with no control on resource availability, service quality and user experience, along with the characteristic cases of the Content Delivery

Networks (CDNs) and the P2P networks. This chapter concludes with the discussion on advances in network virtualised environments.

Chapter 3 “An Architectural Paradigm towards Future Internet” presents the proposal for a content-aware media architecture and its composing environments and layers. The proposed architecture is compared to the FMIA-TT reference model as well as to the most representative Future Internet architectures and particularly to the ones where Media aspects are emphasized.

Chapter 4 “Enabling Content Awareness in Future Media Networks” focuses on the proposed concept of content awareness and the relevant method for traffic/flow identification. In this context, it studies their respective levels of accuracy for a variety of flows and elaborates on the design and implementation of explicit signaling mechanisms (Content Aware Transport Information – CATI). All these are integrated into a prototype network element, the Media Aware Network Element (MANE), able to offer content type recognition and content-based routing/forwarding as a matter of guaranteed QoS/QoE provision in an end-to-end approach. Finally, the chapter presents the experimental results of the forwarding performance evaluation of the MANE.

Chapter 5 “A Network Resource Management Framework in Virtual Content-Aware Infrastructures” motivates the proposed synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification towards efficient resource exploitation: the Intra-Network Resource Management (Intra-NRM). It is THE system responsible for monitoring the network layer resources, operation and status, for controling/configuring the network nodes inside the domain (MANE elements, core routers) and for delivering information regarding the underlying network status and dimensioning to the CAN Manager. Finally, this chapter elaborates on efficient network resources provisioning utilizing existing traffic engineering techniques and mature network technologies for QoS (MPLS, DiffServ), while embedding them in the content awareness framework.

The last chapter of the document, Chapter 6, concludes by summarising the scientific outcomes and research results, besides elaborating on fields for future exploitation.

2. STATE OF THE ART REVIEW AND RELATED WORK

2.1. Introduction

The extensive use of the Internet for the distribution of multimedia content, provided mainly via one-way broadcasting platforms, has led to drastic evolutions in the networked media content value chain, especially due to the open, interactive and multi-service nature of the Internet. The original networked media content value chain composed essentially of the Content Creator, the Content Provider, the Content Distributor (assuring means for delivery through e.g. a broadcasting network) and the End-Users (Content Consumers) had to extend to include existing Internet players, such as the Service Providers (assuming, among others, Content Distributor's role) and the Network Providers (or Operators), in order to be able to handle this evolution. A representation of the current media value chain is depicted in Figure 1.

Currently user-generated media content conquers Internet and is ubiquitously offered over a variety of applications through social media platforms. Consequently, there is clearly a necessity to strengthen the role of networked media content value chain actors so as to be able to respond to this increasing demand of media content and services over the Internet. The inherent overall limitations of the current Internet have been repeatedly identified by recent studies [9] and have been summarised on **the general best-effort behaviour, associated with the content/service agnostic nature of the Internet network layer and the lack of cooperation between transport and higher layers.** Especially with regard to the global provision of multimedia services, many of these weaknesses are quite critical and pose serious obstacles.

Extensive research effort is being conducted to overcome these limitations in Europe, USA, Japan and other countries towards defining Future Internet, either in an evolutionary way or from a clean-slate approach [10]. These research studies have all identified the problem of networked media services distribution and consumption. We will next present the roles of each actor of the networked media value chain and describe the limitations they are experiencing, as well as the solutions they deploy.

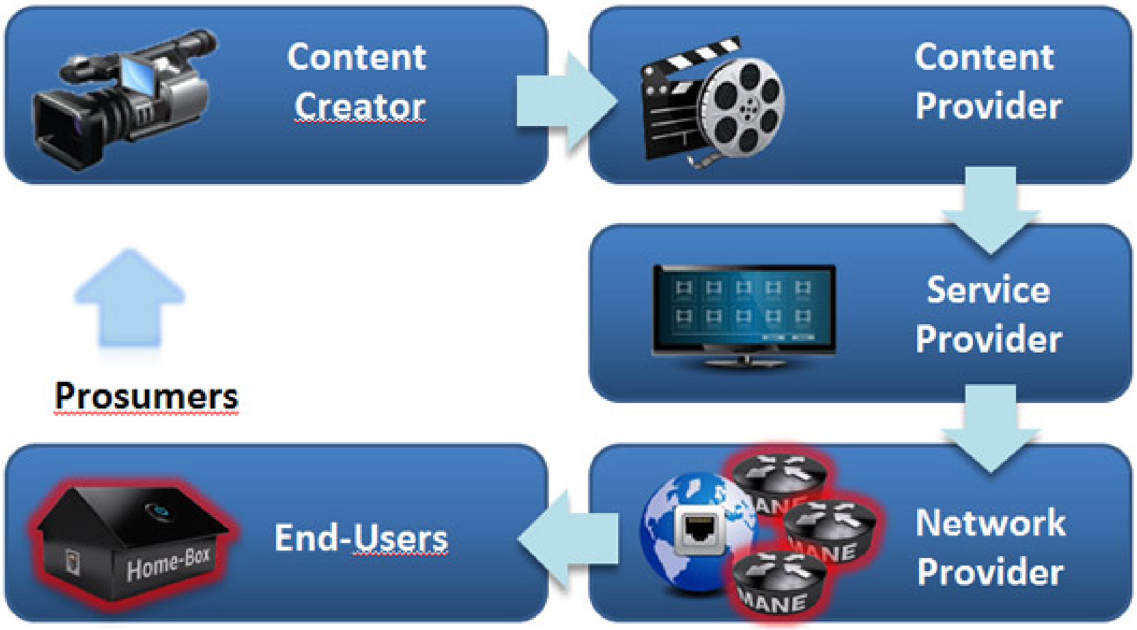


Figure 1 The current Networked Media Content Value Chain - simplified representation

2.2. Current state in the Network Media Content Value Chain

2.2.1. Role of Content Creators

The Content Creator (CC) is an individual or a company who owns the original rights of the content. He may get revenues from selling/renting his content or might just want to publish it for free (social media case). We will distinguish the professional Content Creators (authors, musicians, music/movie companies) from non-professional ones (amateur individuals).

For professional Content Creators (CC), the publication of their content on the Internet is usually handled by the Content Provider(s). Current media services creation platforms, mainly designed for professionals, do not facilitate the creation, sharing, publication and managed delivery of non-professional UGC [11][12][13][14]. Therefore, non-professional CC that wish to act as real Prosumers, serving their content directly from their own equipment, experience serious limitations in terms of service/content creation, publication and dissemination. Indeed, although the Internet is flooded with UGC, the latter is usually provided via centralized content aggregator platforms (such as YouTube, live webcast or social sites), that support limited control and management [15][16][17].

2.2.2. Role of Content Providers

A Content Provider (CP) provides value-added content and applications, by acquiring content from creators and making it available to consumers. A CP can either also act as Service Provider, deploying its own infrastructure for distributing the content, or rely on other Service Providers to distribute it.

The Content Providers want their media content to be distributed with the greatest quality, highest flexibility so that it reaches maximum audience. Today, HD and 3D video are becoming commonplace in distributed content for TV-display devices [18][19][20]. Soon, resolutions are seen to progress to Ultra-HD in 4 and 8K, featuring real 3D captured sound and video for an interactive theatre experience. Such services will require even higher bandwidth, low latency and low loss [21][22][23].

The End-Users are today either obliged to use a specific device to watch streamed content (e.g. IPTV stream with a specific set-top-box) or they have to choose between various proposed formats for consumptions on PCs, laptops, tablets and smartphones [24]. For this purpose, CPs seek unified, simple tools to create new types of content from various CC (and/or locations), enrich, tag and publish them in a very efficient way [25] for ubiquitous consumption. The automatic and simple management, including creation and composition of heterogeneous and various contents, needs amelioration. The CPs' objective is to maximize its profit by always providing the most interesting content to the widest audience with the best quality, while keeping the cost of distribution low.

Due to the proliferation of heterogeneity in networks and devices, most CPs have taken the costly approach of creating series of replications of content into different formats to make them available for different access devices and networks [26]. This approach clearly outlines the limitations experienced by CPs in terms of efficient live adaptation capabilities. Indeed, real-time multimedia applications involve large volumes of data making transcoding a computationally very expensive task [27]. The CPs would expect to have a system where the content would only be stored in one format and automatic adaptation would be done on this format according to (1) the End-User's request and his associated context (preferences, terminal characteristics, location, environment...) and (2) the condition of the delivery path [28].

2.2.3. Role of Service Providers

The Service Provider (SP) provides means and systems to (i) offer (ii) distribute and (iii) manage the distribution of the content (from CPs) and its associated services. To this end, Service Level Agreement (SLA) contracts are usually established between the SP and the CP, when they are independent entities. It is a comon case that Internet-based SPs have difficulties to provide end-to-end QoS at CP and end-users [29]. In an effort to cope with this issue, they have elaborated various solutions, such as:

- Proprietary non-standardized closed solutions with a tight collaboration between the SP and the Network Provider (NP) (provided that only a single underlying NP is involved in content delivery). SLAs are concluded between the CP, SP and NP. The network and the media delivery platform are fully managed on end-to-end basis [30]. Specific software and dedicated devices must be used. This solution is used by

Telecommunications operators having their network deployed, for their own e.g. IPTV/VOD service. Such an example is the Microsoft Mediaroom IPTV Platform [31][32];

- A standardized solution of a fully managed media delivery platform (e.g. ETSI TISPAN IMS-based IPTV Architecture, [33], ETSI TISPAN NGN Integrated IPTV Subsystem [34], Open IPTV Forum [35]). One advantage of such solutions is that they can interact with other subsystems deployed. The software and hardware are compliant with the standard, enabling easier deployment and better inter-operability with external systems. Although the IPTV standardization is ongoing, most of current IPTV services are built on vendor-specific platforms without integration with next-generation network subsystems, usually quite heavy and costly to deploy [36][37]. Nevertheless, none of the existing standards imposes requirements to the Network Provider, leaving as only collaboration between the actors the establishment of SLAs/SLSs (Service Level Agreements – Service Level Specifications) [38];

- A Content Delivery Network (CDN) solution. It consists of a media delivery platform that creates copies of content and places them at strategic locations in the Internet inside dedicated servers, called surrogate servers [39]. Those servers handle the delivery to the End-Users. CDNs use either the active network approach, relying on network components with specific software for identifying content-types or the pure overlay approach, fully independent of network components. The main drawback of this solution is its cost, especially due to the number of surrogate servers needed to be deployed and maintained in order to have efficient QoS/QoE for the End-Users [40]. Additionally, CDNs and all solutions using distributed caching can make the efficiency of distribution better but cannot be extended to meet the Internet scale [41]. Another CDN problem is the lack of cooperation between network and the CDN logical infrastructure. Even in the absence of congestion, the “middle mile” encounters delays as a result of peering problems between transit ISP’s, DoS (Denial of Service) attacks, Link failures, etc. Neither the servers nor the clients have any control over the “middle mile”. This highlights the necessity of a better cooperation between Service Provider and Network Provider, even for this kind of solution [42]. This solution is more and more used by CPs for their WebTV and other Web-based services, such as Apple TV, Netflix, YouTube and Google TV, relying on own infrastructure (Google) or from dedicated companies such as Akamaï;

- A Peer-to-Peer (P2P) solution. P2P-network systems often implement an abstract overlay network, built at Application Layer, on top of the native or physical network topology. Such decentralized systems have a lot of advantages compared to centralized client-server systems [43]. However, the introduction of peer-to-peer networking both increases traffic volumes dramatically in a very short time frame, and changes the structure of this traffic (peer-to-peer traffic is more symetric that the previously dominant client-server model). Besides that, peers must deal with limited and unreliable connectivity, making it hard for network planning [44]. Thus, the provision of a particular QoS/QoE level to the End-User is hardly achievable. As well, P2P application to streaming video technologies is somewhat restricted by some fundamental P2P problems [45][46]: asymmetric bandwidth provisioning (leading to instability when the number of peer-clients for a particular content far outnumbers the peer-servers); selfish behaviour of P2P entities [47]; tussle of interests between P2P networks and ISP’s. The P2P solution is used by several CPs for delivering their WebTV service, such as Veetle, the SopCast, but also Octoshape providing P2P solution for CNN and TVU Player for NBC, Fox News, CBS.

All the aforementioned solutions, in addition to having geographical restrictions and/or deployment limitations, do not assign an active role to the Network Provider, whose only involvement is to offer the connectivity service (with or without QoS assurance, QoS assurance involving coarse requirements in terms of bandwidth, delay, and packet loss but not according to a particular content-type). They are called Over-The-Top (OTT) solutions. By by-passing the Network Provider, such systems do not have the possibility to rely on network information for adapting their content to the variations of the network and problems that may occur. They can only base it on end-points, thus **limiting their adaptation and management features**. Indeed, the management schemes of the solutions above are static and do not involve any feedback from the transport network. Moreover, adaptation according to user context is seldom foreseen, whereas the platform is usually oriented to a single delivery context i.e. fixed wireline reception, mobile consumption, etc.[48][49].

2.2.4. Role of Network Providers

The Network Provider (NP) manages the physical network and IP infrastructure and assures the transport of data and signalling. The Network Operator (NO) is the entity which exploits the network. The NO and the NP can be merged into a single entity. The NP can act as Internet Backbone Provider, providing resources for the backbone of the Internet (Tier 1 and 2), and/or as Internet Service Provider, providing Internet access and transport to the consumer. Today, many Network Providers also undertake the roles of Service Providers (and even Content Providers).

The Network Providers have the difficult role of transporting traffic with important diversity. The current Internet architecture does not provide to NPs sufficient means and tools to be able to cope with the CPs/SPs’ demands for media flows transportation in real time and in an efficient way [50][51][52][53]. Neither the network management nor the network nodes themselves have the **information on the media flows content-types**, or content objects, transported by the packets, making thus impossible dynamic accommodation.

In order to provide an acceptable degree of network-level QoS, NPs apply resource management strategies such as i) Over-provisioning; i.e. accommodating more resources than needed and serving the traffic either in a best effort mode or applying coarse traffic differentiation to traffic aggregates, or ii) Static Provisioning of a given amount of resources (via SP/NP SLAs), optionally supported by traffic differentiation and establishment of virtual paths/trees (e.g. via MPLS – Multi Protocol Label Switching).

In any case, the network usually functions in a **content-agnostic mode**, i.e. applies the same treatment to all flows, regardless of the application which they convey and the requirements of each application. In addition, the NP usually provides the connectivity service “as-is”, **with low dynamicity and also without any feedback.** That is, the SP is not provided with any real-time information on network conditions and usage, which forbids the deployment of network-aware/network-optimised services. Also, applicable to each approach is the underlying still under study **problem of inter-domain crossing** and its impact on cost-effective delivery of media content over multi-domain chains.

2.2.5. Role of End Users

The End-User (EU) is seen as the last element in the Networked Media value chain. The End-Users have always been Content Consumers and from recently are becoming Content Creators as well, especially thanks to user-participating and social applications.

However, the distribution model used by SPs providing these applications still relies on deployed servers, periodically re-dimensioned and replicated to serve the growing mass of potential consumers. As explained on the Content Creators section above, the End-Users are experiencing limited interaction and flexibility with regard to **creation and (mostly) management of their own (basic or composite) services**. Limitations also apply to **security aspects** and to **totally independent and self-managed provision of User-Generated Content and Services**.

In addition to the limitations they face while acting as Content Creators, End-Users are also experiencing limitations as Content Consumers in their own environment. The proliferation of **heterogeneous devices** (tablets, smartphones, laptops, TVs, set-top-box) with various capabilities for service access and content display is leading to the necessity of providing the same content and services via all terminals, with the best possible quality, while at home or away. Today's deployed solutions, such as the ones proposed by telecommunications and mobile operators (web- and mobile portals, home gateways, user profiling, terminals' monitoring agents) still do not fulfil the needs for **media service/content ubiquitous access and context-adapted consumption [54][55]**.

2.3. Current advances in the Content Delivery Architectures

Over the last decades with the increasing Internet growth, driven by the rapid acceptance of broadband access, the content delivery infrastructure is becoming crucial in terms of scalability, reliability and performance. A key challenge lies on delivering more and more complex and personalized contents to rapidly growing end user population. Coping with such an unexpected demand, Service Providers were forced to further exploit delivery-dedicated infrastructures such as Content Delivery Networks (CDN) and Peer-to-Peer (P2P) networks for assuring the delivery of their contents.

2.3.1. Content Delivery Networks (CDN)

A Content Delivery Network (CDN) is a collaborative collection of network elements spanning the Internet, where content is replicated over several mirrored Web-servers in order to perform transparent and effective delivery of content to end users. Collaboration among distributed components can occur over nodes in both homogeneous and heterogeneous environments.

Caching and replication are the key technologies used to improve service performance. Multiple copies of contents are maintained in geographically distributed nodes organized in several clusters. The aim is to bring contents closer to end users. The higher is the distribution, the more scalable and efficient is the content delivery. Indeed, a high distribution strategy means short distances between content servers and consumers and, consequently, minimizes latencies and reduces network bandwidth consumption, as well as possible packet loss occurrences, all these towards a better user experience.

The use of CDNs from SPs seeking reliable and on time content delivery has several advantages such as (a) Offloading the origin servers, (b) Reducing the service/content providers' investments in the delivery infrastructure deployment and management, (c) Bypassing traffic jumps and avoiding peering traffic by bringing the contents closer to end users, (d) Improving content delivery quality, speed and reliability.

Many services, such as web-based services, file transfer services, streaming media services, etc., can take advantage from CDNs infrastructures to deliver their contents. To deliver these services and contents in an efficient way, the design of CDN requires some important features such as replica placement, cache organization, surrogate selection and request routing mechanism. Since the delivery of live and on-demand streaming is more challenging due to the large size of delivered contents and to the long life of the streaming sessions, CDN replica servers should implement additional functionalities such as large and shared caching capabilities, peering capabilities, transcoding capabilities and streaming session handoffs.

2.3.2. P2P Networks

The term Peer-to-Peer (P2P) refers to any relationship between a number of autonomous devices behaving as a peer, i.e., a client and a server at the same time. The P2P networks are networks in which each computer can act either as client (requesting information), or server (offering data) or/and as a combination of client and server, allowing partition of network resources and services, such as information, files, cycle processing and storage, through direct communication between the systems (without necessarily requiring the use of centralized servers). Unlike client-server types of networks, P2P systems promise improved scalability, lower ownership costs, and decentralized coordination of used or restricted resources by exploiting efficient bandwidth monitoring and routing decision mechanisms at the application layer.

2.3.2.1. Centralized Systems

In centralized P2P systems, network topology is controlled and content queries are appropriately handled by pre-assigned overlay entities with specific organization rules. In this model peers communicate with a central server (indexer) in order to publish the content they provide. Upon a request of a peer to the central server the peer receives the content from the best peer(s), taking into consideration the availability of the peer(s) offering the content and other metrics specified by the user. The most common type of centralized P2P networks is the Distributed Hash Table (DHT), resulting in the efficient location of rare content with bounded searching complexity.

2.3.2.2. Decentralized Systems

Following the legal problems of Napster, many file sharing networks shifted to fully decentralized approaches. The decentralized systems use only peers without the need of a server. Each peer node takes functions of both index servers, seeking local resources and routers, connecting directly with a number of other nodes. All search requests and their responses are transferred from each node in the neighboring node. A new node is connected to an existing node. Two separate decentralized systems may be connected together using as bridge a single node. Gnutella [56] and Freenet [57] are examples of such systems. The Gnutella is one of the most popular file sharing networks while Freenet is a distributed data warehouse trying to provide freedom of speech through strong anonymity. The most important advantage of the decentralized systems compared to the centralized ones relies in the fact that the single point of failure does not exist anymore. The important disadvantage is the fact that the transfer of any request by any node to the neighborhood brings large seek times and creates a high volume of network traffic (since all the messages are repeated).

P2P systems can be considered as the logical extreme of the distributed approach for content delivery and, accordingly, are highly scalable. However, if the P2P systems represent a promising low cost approach for highly scalable video content distribution, they also present some weaknesses such as lack of control, high peer churn and unfairness and significant imbalance between the uplink and downlink capacity. These weaknesses may rapidly result on system saturation and poor quality.

2.4. Current advances in Virtualized Environments

2.4.1. Network Virtualization

Network Virtualization has been presented by the research community as a key enabler technology to escape from the current well-known limitations of the Internet. Moreover, it is also seen as a sustainable tool for experimenting novel network protocols on production networks without affecting other critical services, running of the same substrate network. It is widely proposed to be an integral part of the Future Internet.

In the past years, network virtualization has received significant attention, as surveyed in [58]. Future Internet projects, such as 4WARD [59], Cabernet [60], GEYSERS [61], presented network virtualization architectures with emphasis on the business roles and the interfaces required for the provisioning and management of virtual networks across multiple domains. [62][63] presented early prototype implementations which realize several components of the 4WARD network virtualization architecture, while their work continued in [64][65] showed that this architecture is technically feasible and robust. Several platforms have been deployed, assisting network operators to deploy virtual networks on their own infrastructure [66].

In terms of virtual network embedding, most deployed algorithms [67][68] consider a single substrate provider and require full knowledge of the available resources and the underlying network topology. Recent work [69] presents a multi-domain virtual-network (VN) embedding framework. This approach consists in relaying VN requests across infrastructure providers till the embedding has been completed. However, this VN embedding approach lacks of algorithms for resource assignment and allocation and it has not been evaluated. Hence, it is unclear how fast it converges to a complete VN embedding. [70] provides a set of algorithms for multi-domain VN embedding. According to [71], resource planning becomes more complicated if computing constraints on the network elements are also to be taken into account. Current works focus on designing a heuristic network embedding (service mapping) algorithm addressing joint in-network link and computing assets which will be efficient, yet reasonably complex i.e. achieving an acceptable trade-off between computational time and embedding optimization.

With regard to node virtualization, advances on server (e.g., [72]) and link (e.g., [73][74]) virtualization provide the technological ingredients needed to deploy virtual networks at global scale. In addition, [75] showed that virtual routers on commodity hardware have the capability to forward minimum-sized packets at several Gbps, while offering a high level of programmability [76]. Platforms, such as VINI [77] and Trellis [78], synthesize server and link virtualization technologies to build simple virtual networks, mainly used for experimentation. In most cases, a virtual router provides an illusion of isolation, rather than a real isolation, as it lacks dedicated memory, processing and I/O resources [79]. Currently, many researchers work on implementing cloud-based mechanisms for deployment of virtualized network appliances as isolated Virtual Machines (VMs) with fully managed resources into clusters of commodity computing nodes. The aim is to effectively address one of the main open issues in the area of network virtualization: the need for dynamic resizing of allocated in-network resources (up- and down-scaling of VMs according to function requirements and traffic load) and also for handling failures by migrating Virtual Routers VRs to another physical location.

2.4.2. Software Defined Networking (SDN)

Network programmability is a promising means to overcome network technology ossification by enabling the rapid deployment of new protocols and functions into deployed operational networks. The most popular paradigm for vendor-neutral network programmability, i.e. Software Defined Networking (SDN) is a model for network control, based on the idea that the handling traffic flows can be made programmable at scale, thus enabling new dynamic models for traffic management. SDN provides abstraction at three areas of the network: distributed state, forwarding and configuration [80]. SDN is an approach to networks that enables applications to converse with and manipulate the control software of network devices and resources. SDNs comprise applications, control software, and interfaces to services that are hosted in an overlay or logical/virtual network as well as those possibly same components that comprise the underlying physical network

2.4.2.1. SDN against Current Network Limitations

Some of the current networking technologies limitations that SDN hopes to solve are summarized below.

a. **Complexity that leads to stasis:** today, networking consists of *many discrete sets of protocols* (connecting hosts over arbitrary distances, link speeds, and topologies) defined in isolation, to solve specific problems with no benefit of any fundamental abstractions. This creates complexity: to add or move any device, an IT admin must (re-)configure multiple HW/SW entities using device-level management tools. He should consider topology, vendor switch model, SW version, etc.; that is why today's networks are relatively static, as admin seeks to minimize the risk of service disruption. The static nature of networks is not good for today's dynamic server environment, (server virtualization, VMs migration) and offers limited capability for dynamic QoS differentiation levels (low capability to dynamically adapt to changing traffic, application, and user demands).

b. **Inconsistent policies:** to implement some network-wide policy, one needs to configure many network devices and mechanisms. The complexity of today's networks makes it very difficult to apply a consistent set of access, security, QoS, and other policies.

c. **Scalability issues:** in today complex networks (including big data centers) over-subscription based on predictable mid-long term traffic patterns is not working well; traffic patterns are highly dynamic and therefore unpredictable on long term. Carriers have to deliver better-differentiated services to customers. The network must serve very large groups of users with different applications and needs.

d. **Vendor dependency:** carriers/enterprises want rapid response to changing business needs or user demands, but their ability to respond is limited by vendors' equipment product cycles (years). The lack of standard and open interfaces limits the ability of network operators to tailor the network to their individual environments.

e. **Changing traffic patterns:** these have changed significantly especially within the enterprises data centers. Today's applications access different databases and servers, creating a high Machine-to-Machine traffic before returning data to the end user device (different from classic client-server applications). Also, users-network traffic patterns are changing: they want access to corporate content and applications and with various QoS, from any type of device, anywhere, at any time.

f. **Cloud services development:** there is a significant growth of public and private cloud services (SaaS, PaaS, IaaS, NaaS, …) on demand and "à la carte"; IT's needs for cloud services are various: security, compliance, auditing requirements, elastic scaling of computing, storage, network resources, etc.

g. **Need for more bandwidth:** there is an increasing need of high bandwidth, especially for content-media related services and communications.

2.4.2.2. SDN architecture approach

The major SDN [81] characteristics are that:

1. The *Control Plane (CPl)* and *Data Planes (DPl)* are clearly decoupled;
2. Communication between them is accomplished by a new vertical protocol (e.g. OpenFlow [82][83]);
3. Network intelligence and state are logically centralized;
4. Underlying network infrastructure is abstracted from the applications.

The SDN technology brings high promises to enterprises and carriers: higher programmability opportunities, automation, and network control; enabling them to build highly scalable, flexible networks, including fast adaptation to changing business needs.

Regarding SDN and OpenFlow, the research community has identified the following advantages [84]:

- H*igh-performance, granular traffic control* across multiple vendors' network devices with the ability to apply comprehensive and wide-ranging policies at the session, user, device, and application levels;
- *Centralized management and control* of networking devices improving automation and management;
- *Common APIs abstracting the underlying networking* details from the orchestration and provisioning systems and applications;
- *Flexibility in terms of* new network capabilities and services with no need to configure individual devices or wait for vendor releases;
- *Programmability* by operators, enterprises, independent software vendors, and users (not just equipment manufacturers) using common programming environments;
- *Increased network reliability* and *security* as a result of centralized and automated management of network devices, uniform policy enforcement, and fewer configuration errors;
- *Better end-user experience* as applications exploit centralized network state information to seamlessly adapt network behavior to users' needs;
- *Protects existing investments* while future-proofing the network;
- *With SDN, today's static network can evolve into an extensible service delivery platform capable of responding rapidly to changing business, end-user, and market needs.*

2.4.2.3. Software Defined Internet Architecture

Very recently, the SDN concepts have been proposed to be extended in so called *Software Defined Internet Architecture* (SDIA), [85][86][87]. The motivation is shortly presented below.

In current networks, a coupling exists between architecture and infrastructure: a given architecture cannot be supported on any infrastructure. If architectural changes are wanted, then significant costs are involved for development (manufacturers) and deployment (operators). That is why a decoupled architecture is advocated, based on SDN approach and other principles, e.g. as separation between the core networks and edges. It is known that incremental attempts to address Internet architectural deficiencies had limited success. *On the other side, none of the very new, clean slate architectures have succeeded in being deployed.* It is observed that currently a main obstacle to the flexibility consists in coupling architecture (mainly protocols) and infrastructure (routers, switches, transmission systems, etc.): any significant change to IP (or its successors) would need to replace (or overhauling) the routers, because the forwarding ASICs only have limited flexibility.

In [88], an SDIA proposal is done based on ideas taken form SDN, MPLS (separation core-edge), Middleboxes and Software forwarding. The idea is to use these features in a synergetic way: the data plane can consist of a network *core*, using its own internal addressing scheme (similar to L2 networks of today), and a network *edge* that uses software forwarding. All architectural dependencies reside at the edge; but these can be easily modified because there, packet forwarding is done in “software”. The control plane uses SDN to control the edge routers (to specify their forwarding actions), and leaves the design of the core control plane up to each domain. Any protocol similar to OpenFlow could be used.

2.5. Summary

In this chapter, we have provided a brief overview of networked media chain and the roles of the main actors. We have also described the current advances in the main content delivery architectures. In particular, CDN and P2P technologies are currently used to distribute media contents over Internet. For each solution, we have presented the basic concepts on which it is built, its advantages, drawbacks and limitations. None of these delivery solutions take into account the Content Awareness feature that is critical for identifying content in transit and mapping its QoS/QoE requirements into specific network characteristics. Moreover, we presented current trends in areas of Network Virtualisation and SDN as they are considered key features of future network infrastructures, enabling increased flexibility and collaboration among network and service providers. In this thesis we will focus on virtualization techniques creating parallel content-aware virtual planes later introduced as virtual content-aware networks. The following chapter will present a content-aware future media Internet architecture that proposes an integrated service and network management solution, which is both service- and user-centric.

3. AN ARCHITECTURAL PARADIGM TOWARDS FUTURE INTERNET

3.1. Introduction

The widespread adoption of telecommunications, Internet and broadcasting technologies in citizens' day-life has led to the emergence of a plethora of e-services/applications, rich in user-generated/centric content, with high demands for network resources and stressed QoE requirements in an end-to-end approach. For example, Content Distribution, Digital Interactive TV, Video/Audio on Demand, Live streaming, and Social Networking are some representative cases intrinsically characterized by the active user participation both in the media generation and delivery processes, which raise the network resource requirements and creating, therefore, new challenges in network operations, configuration and administration. Promising solutions towards alleviating these issues are a) Content/Service Awareness at the infrastructure level, i.e. the network capability to analyze media-flows and adjust its operation according to the content/service requirements for network available resources and the requested QoS/QoE policies, and b) Network Awareness at the service level, i.e. the capability of service/content adaptation according to the underlying network infrastructure and the requested QoS/QoE policies.

3.2. The ALICANTE approach

3.2.1. The high-level architecture overview

Building upon these two state-of-the-art approaches/solutions, ALICANTE[1] project introduces a novel architecture (as depicted in Figure 2), utilizing both Service and Network Awareness, besides moving one step beyond by converging them into a common operational system. The unifying factor in this consolidation process is Context Awareness, i.e. the ability to dynamically adapt Network and Services to each other, according to the users' operational environment and the requested QoS/QoE policies. Towards this, ALICANTE elaborates on two novel virtual layers on top of the traditional network layer, namely the Content-Aware Network layer (CAN) for network packet transport and processing, and b) the Home-Box layer (HB) for the actual content adaptation and delivery. Seamless integration and efficient collaboration among them is achieved via novel communication interfaces, enabling user-centric networked media, highly heterogeneous in terms of user functions and distributed in the network, to be enrolled in the frame designated by the operations of traditional networks and thus be made

[1] EU FP7 ALICANTE project: http://www.ict-alicante.eu/

widely available to the users with adequate maximum possible QoS/QoE along with optimum resource exploitation [89].

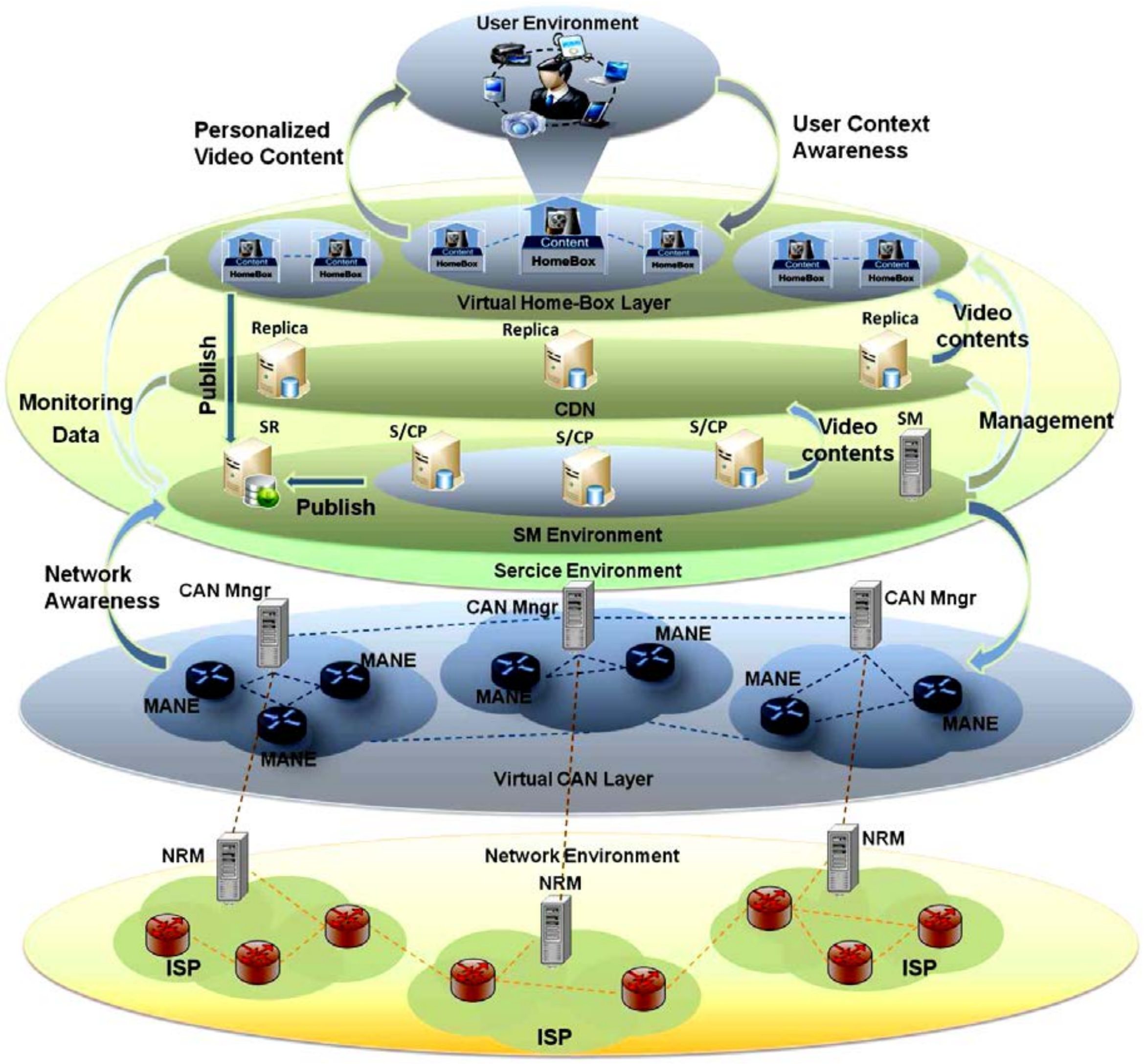


Figure 2 ALICANTE High level Architecture Overview

More specifically, the CAN layer constitutes an overlay network infrastructure, composed of Virtual Content-Aware Networks (VCAN), spanning single or multiple domains. The VCAN aims at a) improving data delivery by classifying and controlling messages in terms of content, application and individual subscribers and b) providing QoS assurance, by classifying the packets and associating them to the appropriate virtual path. Major component in the CAN layer is the Media-Aware Network Element (MANE), providing (among the others) in-network adaptation of media flows. On the other hand, the HB layer constitutes of a mesh of media-centric home gateways (i.e., the Home-Boxes), which resides in the end users' premises, capable of enabling uniform access to the multimedia content for heterogeneous terminals. At this layer, advanced management and monitoring functions provide real-time information about user context and network conditions, allowing better control over content delivery, as well as intelligent adaptation.

3.2.2. Overall system architecture

Below is presented the design of the **ALICANTE** overall macroscopic functional architecture (subsystems, functional blocks and interfaces). The functional architecture comprises of subsystems, high-level blocks and interfaces, which constitute an abstraction of the specific centralised/distributed characteristics.

Figure 3 details the five **ALICANTE** architectural subsystems (End-User Terminal, SP/CP, Home-Box, CAN and Network subsystems) by presenting the main functional blocks of each subsystem as well as the main horizontal and vertical interfaces between them. The latter have been labelled using the initials of the subsystems they are interconnecting:

T: End-User Terminal Subsystem
S: SP/CP Subsystem
H: Home-Box Subsystem
C: CAN Subsystem
N: Network Subsystem

and of the function they serve:

s: Signalling-Control
m: Monitoring
a: Adaptation

The rule applied is XYz, where **X** is the upper layer subsystem, **Y** is the lower layer Subsystem and **z** is the functionality supported. For example CNm denotes the interface conveying *monitoring* data between the *CAN* and *Network* subsystems. The interfaces indicated may represent either single interconnection or a bundle of interfaces between two modules, serving multiple purposes. More details about the interfaces can be found in [151].

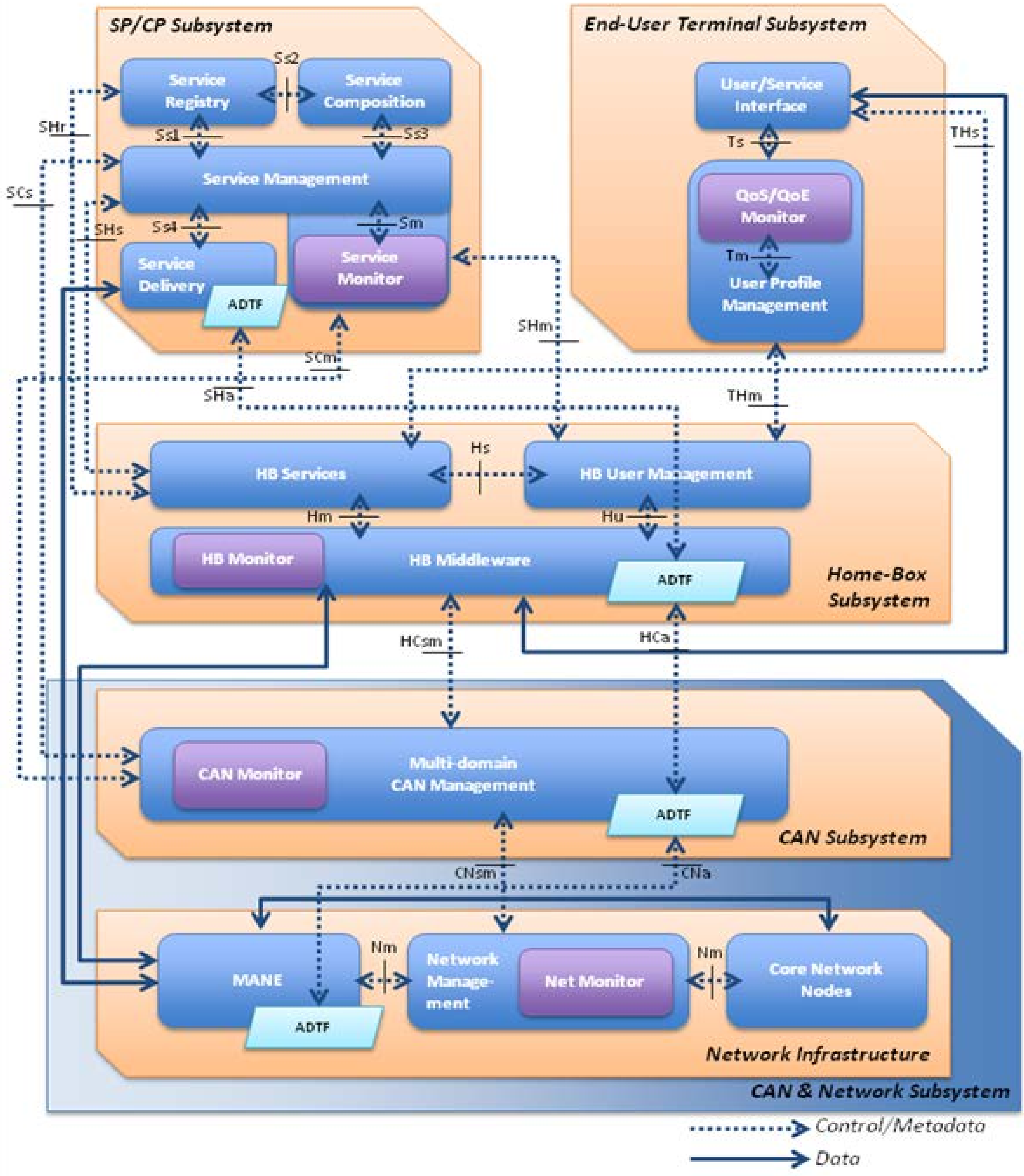


Figure 3 High-level functional architecture of the ALICANTE system

3.2.3. ALICANTE architecture and Future Internet standardization efforts

The Future Internet is an emerging area where already a lot of on-going work is done. Through its Internet Architecture Task Force, FIArch has highlighted the main limitations of the current Internet architecture, which the Future Internet design should take into account [99]. A detailed up-to-date FIArch work can be found in [98]. New Internet architecture models are proposed in [90][91][92][93][94][95].

The Future Internet is expected to be a communication and delivery ecosystem. The Future Media Internet – Think Tank group (FMI-TT) aims to specify a reference model of a "Future Media Internet Architecture" which covers delivery, adaptation/enrichment and consumption of media within the Future Internet ecosystem and has defined a high-level FMI network architecture based on four macro-layers (or strata) as depicted in Figure 4.

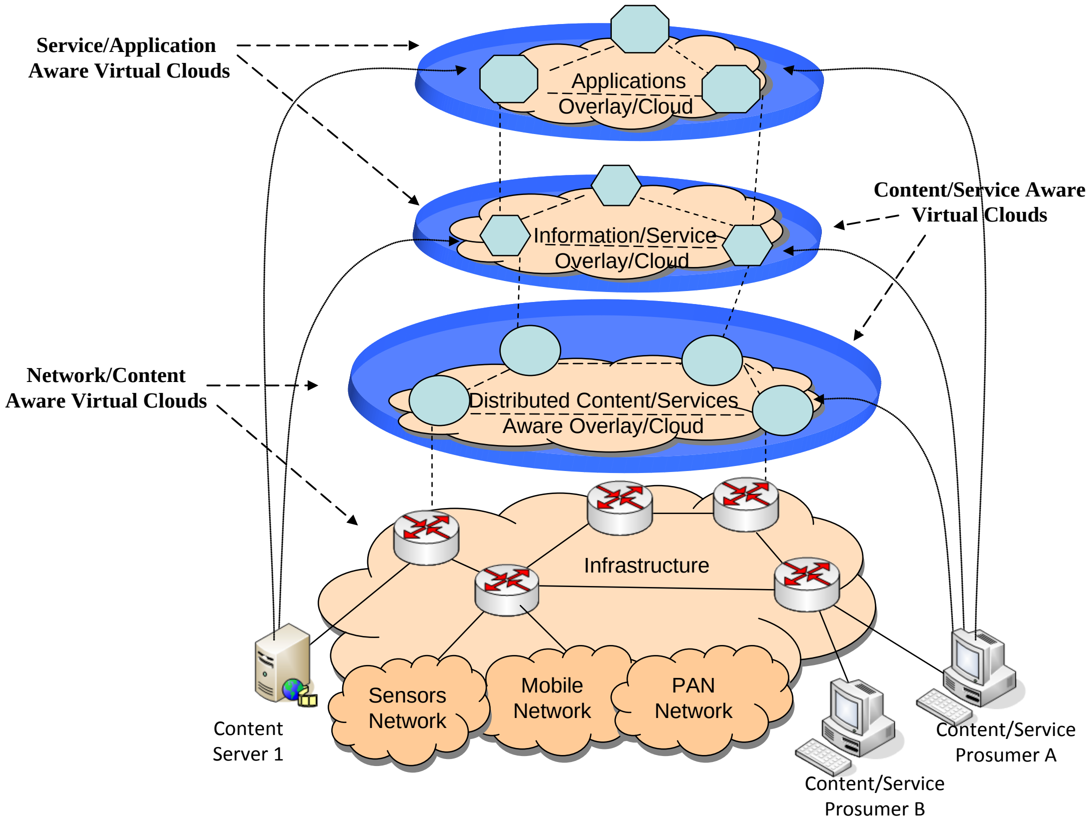


Figure 4 FMIA-TT Reference Model

a. Applications Overlay (AO): it includes applications which may use different services and the information delivered by the Information Overlay (IO) and the media/content themselves;
b. Information Overlay (IO): it comprises intelligent nodes and servers with knowledge of the content/Web-service location/caching and the network instantiation/conditions (the nodes can vary from P2P peers, secure routers or even Data Centres);
c. Distributed Content/Service-Aware Overlay (DCSAO): it contains the CAN nodes which filter content and Web services;

d. Service/Network Provider Infrastructure (SNI): it can be seen as the traditional layer of services offered by the ISPs. The users can be providers and/or consumers ("prosumers") of the services offered by this layer.

ALICANTE architecture can be seen as a solution deriving from the FMIA-TT reference model. Even though the mapping between them is not exactly one-to-one since ALICANTE does not have in its scope all FMIA-TT functions, the correlations are the ones presented in Table 1.

Table 1. FMIA-TT and ALICANTE Architectures Mapping

FMIA-TT Architecture	ALICANTE Architecture
SNI	Network Environment+Service Environment
DSCAO	CAN Layer
IO	HB Layer
AL	User Environment + Service Environment

3.2.4. ALICANTE compared to other existing Future Internet architectures

A significant trend recognized in Future Internet is the information-centric orientation and in particular, content orientation. Consequently, new concepts (partially overlapping) and architecture proposals like Information-Centric Networking (ICN), Content-Oriented/Centric/Aware Networking (CON/CCN/CAN), have emerged, proposing significant changes in communications and networking, including those related to basic architectural principles. One main proposal in ICN/CON is to replace the conventional host identification approach (or, more general, "host-centric" paradigm) with content identification (i.e. "information centric" or "content-centric"). While ICN/CON/CCN approaches are very promising, they raise many challenges like the degree of preservation of the classic TCP/IP layering principles, naming, addressing, routing, forwarding, and management .

This section briefly describes current and past research efforts, that address a subset of the challenges for a Future Media Internet architecture.

The FP6 project **MESCAL** "Management of End-to-end Quality of Service Across the Internet at Large" project, [100][101] proposed an evolutionary, scalable, incrementally deployable architecture, enabling flexible deployment and delivery of inter-domain QoS across the Internet at large. The MESCAL business model actors are: Service Providers (SPs), IP Network Providers (INPs), Physical Connectivity Providers (PCPs) and Customers. MESCAL developed a generic, multi-domain, multi-service functional architecture, and a complex management system mainly focused on resource management and traffic engineering (offline and online) intra and inter-domain. However, several MESCAL limitations exist with respect to **ALICANTE**: the End-Users cannot act as content providers; it is focused mainly on networking aspects and does not consider the service; it does not have a multimedia orientation as a main design direction. Content-aware networking, P2P mode, HBs are missing. While **ALICANTE** may use the MESCAL concepts of QoS classes (local, extended, meta-QC) in a multi-domain environment, it brings additional features mentioned above, which are missing in MESCAL.

The FP7 project **4WARD** "Architecture and design for the future Internet" [91], is a large research project clearly oriented towards FI. It proposes new Architecture Concepts and Principles (NewACP) based on a plurality and multitude of network architectures: the best network for each task, each device, each customer, and each technology. Networks coexist and complement each other, each of them addressing individual requirements such as mobility, QoS, security, resilience, wireless transport and energy-awareness. The business players are: Physical Infrastructure Provider (PIP), Virtual Network Provider (VNP) (assembling virtual resources from one or multiple PIPs into a virtual topology), Virtual Network Operator (VNO) (installation/operation of a VNet over the virtual topology provided by the VNP for a tailored connectivity service), Service Providers (SPs) (use the virtual network as a support for their services - these can be value-added services and then SPs act as application service providers, or transport services and then SPs act as network service providers). Full network virtualisation is considered in 4WARD to solve the interoperability and is considered a main concept for a clean slate FI approach. **ALICANTE** benefits from the virtualisation aspects investigated in 4WARD, however, it aims at an evolutionary, backwards-compatible approach rather than a clean-slate one, in order to maximise adoption possibilities and accelerate market penetration.

The FP7 project **COMET** "COntent Mediator architecture for content-aware nETworks", [92][93] aims to provide a unified interface for content access whatever the content characteristics are: temporal nature (pre-recorded or live), physical location (centralised or distributed), interactivity requirements (elastic or real-time), or any other relevant features. It also aims to apply the most appropriate end-to-end transport strategy: by mapping the content according to its requirements and user preferences to the appropriate network resources; best quality of experience for End-Users; it supports unicast, anycast and multicast. All these are achieved while preserving network availability and structural resilience, as key factors in perceived QoE. The business entities are: Content Consumers (CC), Content Providers (CP), COMET-capable ISPs and carriers. COMET aims to be a flexible framework to accommodate several possible current or future content-related business models. COMET distinguishes two kinds of scenarios: free content access and charged content access. It is identified that COMET and **ALICANTE** have overlapping domains. However, the COMET business model is only partially sufficient for **ALICANTE** needs; it does not consider fully the cooperation between network overlay and network resources, but is focused mainly on mediation activities.

In [105] , a revolutionary solution is proposed as *Content-Centric Networking (CCN).* The idea stems from the fact that the IP networks are increasingly used for content distribution and retrieval, while networking technology still are based of connections between hosts. CCN replaces the traditional "where" paradigm used for IP routing with "what"-identifying the content by taking the content as a primitive – decoupling location from identity, security and access, and retrieving content by name. Using new approaches to routing named content, derived heavily from IP, the study claims that one can achieve simultaneously scalability, security and performance improvements. While not applying full CCN concepts, the **ALICANTE** solution is capable to be incrementally developed towards a CCN a solution, given its content awareness incorporated in the **ALICANTE** edge routers.

In a recent work [95], a mixed content centric solution is proposed. The ***CURLING*** architecture, "*Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services*", aims at efficiently diffusing media content of massive scale. It entails a holistic approach, supporting content publication, resolution and, content delivery and provides to both CPs and customers high flexibility in expressing their location preferences when publishing and requesting content, respectively, through *scoping* and *filtering* functions. Content manipulation operations can be driven by factors as business relationships between ISPs, local ISP policies, and specific CP and customer preferences. Content resolution is also coupled with optimized content routing techniques that enable efficient unicast and multicast-based content delivery across the global Internet. **ALICANTE** and CURLING are complementary. **ALICANTE** is less content-centric (content-type based only) in the control plane, but more powerful in assuring efficient media flow QoS enabled transport based on content awareness.

The FP7 project ***OCEAN*** "Open ContEnt Aware Networks" [106] , designs a new open content delivery framework that optimizes the overall QoE to End-Users by caching content closer to the user than traditional CDNs do and by deploying network-controlled, scalable and adaptive content delivery technique. OCEAN aims to find solutions to the imminent problem of multimedia content traffic clogging up the future aggregation networks, when the offering of online video of high quality over the Open Internet continues to increase. OCEAN builds innovative self-learning caching algorithms that meet the specifics of the highly unpredictable location and time-dependent consumption patterns and dynamically adapt to the rising popularity of future delivery services. Media-aware congestion control mechanisms based on slight, but controlled quality degradation is suggested than blocking of user requests. OCEAN and **ALICANTE** are complementary, in the sense that the first addresses content-awareness from the caching point of view, whereas the second mostly studies traffic differentiation and scheduling issues.

To summarize, Table 2 presents a main features comparison between the aims of **ALICANTE** and the projects presented above. This table does not aim at a direct competitive comparison; instead, it lists all aspects related to Media Ecosystems deployment, which are partially covered by the aforementioned projects and fulfilled within **ALICANTE**.

Table 2. Comparison of ALICANTE with other FI oriented projects

Main Features	MESCAL	4WARD	OCEAN	COMET/ Curling	ALICANTE
Media (including real-time) services oriented	No	No	Yes	Yes	Yes
Full high-level services management	No	Yes	No	Yes	Yes
Multi-domain	Yes	Yes	No	No	Yes
Network Virtualisation	No	Yes	No	No	Yes
CAN/NAA concepts	No	No	Yes	Yes	Yes
Network Resource Management	Yes	Yes	Yes	Yes	Yes
P2P capabilities	No	No	No	No	Yes

3.3. Summary

The ALICANTE concept contributes to the largely agreed and fundamental key statement that the Future Internet will be mainly content and services oriented. Therefore, ALICANTE proposes an integrated service and network management solution, which is both service- and user-centric. The proposed architecture relies on two main pillars: first, an innovative Service Environment, involving both a complex Service Management, complemented by an overlay of interconnected media-centric Home Gateways ("Home-Boxes") and second, a radically enhanced Network Environment, featuring inherent Content Awareness, enabling the creation of flexible virtual overlays networks dedicated to media transport. At the same time, it assures seamless deployment and transition from the current Internet to future Media Ecosystems, by accepting non-discriminated traditional Internet traffic. On top of the Service Environment, a flexible User Environment is established to provide ubiquitous service access in various usage scenarios a user profiling solution and real-time Quality of Experience monitoring.

This PhD thesis focuses on enabling content aware capabilities at the network layer, i.e., represented by the bottom part of ALICANTE architecture. Whereas the architecture described in this chapter has been proposed as a collaborative common work between a large number of partners, the following two chapters of contributions will detail the proposals made within the scope of the thesis, improve and incorporated within the scope of the ALICANTE project.

4. MANE: A CONTENT-AWARENESS ENABLER IN FUTURE MEDIA NETWORKS

4.1. Introduction

The demand for accurate traffic identification has become over the past years one of the major research challenges since network operators and administrators need to be promptly informed about the traffic type that is flowing over their networks in order to effectively enhance their infrastructures in terms of capacity planning, traffic engineering, application performance, QoS, intrusion detection and filtering of internet content [111].

The earliest and simplest approach to traffic identification is the port-based method which is focused on examining the port numbers in TCP or UDP headers [112]. This solution of the well-known ports classifies the traffic according to the ports registered in the IANA [113]. However, many applications are increasingly using unpredictable and obscure port numbers. Consequently, the port-based method arises many issues.

In order to deal with the disadvantages of the above method and totally avoid reliance on the semantics of port numbers, payload-based identification method is proposed to inspect the packet payload [114]. Several payload-based analysis techniques have been proposed to inspect the packets' payload searching for specific information strings. Deep packet Inspection (DPI) focuses on analyzing all the content of data packets passing through the network, which includes the headers and the data protocol structures and compares this content against rules, defined as signatures. Signatures are a sequence of textual characters or numeric values with the same pattern chosen for uniquely identifying an associated application (or protocol) [115].

An additional approach for traffic identification would be an approach that is based on host behavior. This technique aims to discover behavioral patterns of the hosts and the services offered by the hosts through the interactions between them [116]. The inherent behavior of the hosts is attempted to be captured in three levels. At a social level, where the behavior of a host is characterized by its communication with other hosts; at a functional level, where it is analyzed if a host is provider or consumer; at an application level, where the origin of the application is identified via port determination.

Payload-based inspection methods show in some cases diminishing effectiveness, since they impose significant complexity and processing load on the traffic identification device. The fact that it is necessary to maintain thorough and up-to-date knowledge of the protocol semantics arises many issues especially when dealing with proprietary protocols or encrypted traffic. Furthermore, inspection of the session and application content may violate privacy policies and legislation.

Newer approaches that rely on traffic's statistical characteristics have been proposed. Traffic statistical attributes such as distribution of flow duration, flow idle time, packet inter-arrival time and packet lengths are considered to uniquely distinguish certain classes of applications, with ultimate goal to cluster IP traffic into groups that acquire similar traffic patterns.

Significant effort has been put on data mining techniques and machine learning algorithms using flow features for traffic identification. Machine learning technique is pioneering tool that aims to classify data based on either a priori knowledge or statistical information extracted from specific datasets. This method can be well suited with Internet traffic identification, as long as the traffic can be classified into categories that exhibit similar characteristics in parameters. Machine learning algorithms are generally divided into supervised learning and unsupervised learning. Supervised learning requires training data to be labeled in advance and produces a model that fits the training data. Unsupervised learning principally clusters flows with similar characteristics together. It does not require training and new applications can be classified by examining known application in the same cluster [118][119][120][121][122].

Due to experimental nature of statistical techniques and their current limitations, the mostly used technology for traffic identification is still DPI.

4.2. Deep Packet Inspection (DPI)

The Deep Packet Inspection (DPI) approach suggests that network flows are inspected and information is extracted from higher layers of data packets (up to the application layer). The exploitation of DPI methods for traffic classification is built around two basic assumptions:

- Third parties unaffiliated with either source or recipient are able to inspect each IP packet's payload (i.e., is the payload visible?);
- The classifier knows the relevant syntax of each application's packet payloads (i.e., can the payload be interpreted?).

The first assumption has created a lot of debate, due to the Network Neutrality issues. In our case, the proposed DPI algorithm only uses an indicative, small number of the initial packets from each flow in order to identify the content. This is achieved through the exploitation of flow-awareness functionality that will allow the DPI process not to classify every single packet by a flow.

For the second assumption, a library of protocol signatures and filter strings has to be built. This library will be referenced in order to detect the protocol. Depending on how this library is built, the detection can be arbitrarily accurate. However, the larger it gets, the more resources are needed for the identification procedure. The signature library needs to be constantly updated as application protocols evolve or as new protocols emerge. Relying on an out-dated library would have a severe impact on the accuracy of the identification

An inevitable issue, arising from the nature of DPI, is that some protocols or traffic flows encrypt the content or even some upper-layer protocol headers. In these cases, it is evident that DPI cannot be an applicable solution for classifications. The proposal of decrypting the encrypted part, applying DPI and then re-encrypting, is not applicable as it raises many architectural and performance-related issues. To overcome the challenge of classifying encrypted data, alternate methods are proposed:

The common approaches used in DPI for identification can be summarized as:

- *Automation* – Tracks partially matched patterns in the text by transition in either a Deterministic Finite Automaton (DFA) or Non-DFA implementation that accepts strings in the pattern set. The main drawback of his approach is that the memory space needed to keep the patterns is big in expense to the lower time complexity (Non-DFA) or the opposite for the DFA method. However, it is the most widely used method for pattern matching;

- *Heuristics* – During the search, a search window of *m* characters covers the text under inspection and slides throughout the text. A heuristic can check a block of characters in the window suffix for its appearance in the patterns. It determines whether a match occurs and moves to the next window position if not. This approach provides the ability to skip characters not in a match, in order to accelerate the search. An example application of this approach is the classifications of packets carrying HTTP get command. An HTTP get request always contains a string similar to the following: "GET /images/logo.png HTTP/1.1". So, a pattern like "GET * HTTP" can be a very effective way of detecting HTTP requests among network packets;

- *Filtering Based* – This approach searches the text for necessary pattern features and quickly excludes the content not containing those features. A very common way of applying this approach for text filtering is using well-known Bloom filters. This method is useful for the extraction of substrings from regular expressions, and filtering text with them or checking for different pattern lengths.

In this thesis, all three approaches were studied, taking into account the performance issues as well as the resources consumed by each method. To enhance the range of services and the accuracy of DPI, numerical properties of the packets can be used. Properties such as packet sizes, number and size of response packets to requests are a factor that can help in the service detection procedure.

An important design aspect is that the protocols that are required to be matched will be limited to those exploited by **ALICANTE** services, thus narrowing the matching database. This differentiates the use of DPI in the frame of **ALICANTE** from the use of DPI for security related and content filtering software that have a much broader scope, needed to address many and different protocols and content-types. Moreover, and in the case where **ALICANTE**-compatible media service providers are considered, the framework simplifies the traffic identification process via explicit signaling of content type that is applied over an extraneous bit-field namely the **Content Aware Transport Information (CATI),** which is injected from service generation nodes in selected places inside application layer protocol headers.

4.3. Content-Aware Transport Signalling

The CATI is the solution proposed as a result of the studies on Content-Aware signaling techniques. It is a signaling structure, inserted by all types of content servers (i.e., coming from the Content Providers or from the End-Users, through their Home-Boxes, in the context of ALICANTE Future Internet proposed architecture) in order to realize the Content Awareness of the Network Environment. The media packets are marked so that the ingress network routers are able to treat the flows accordingly. CATI Format details are described in Table 3 and Table 4.

Table 3. CATI header structure

0										1										2			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
U	A	M	E	STYPE				SST			SPR		VCANID						RES				

The CATI header consists of 24 bits and is inserted either in the RTP or HTTP extension headers, depending on the service. Via the "STYPE" and "SST" fields, the Service Provider can explicitly signal the Service Type and Sub-Service Type to the Network Provider. The network routers involved in the delivery of the stream parse this header and apply the corresponding policies to the media flow, as have been agreed in the Service Level Agreement between the Service Provider and the Network Operator.

Table 4. Content-Aware Transport Information (CATI) and semantics.

FIELD	Definition	Values	Justification
U	Multicast or unicast service	0: unicast 1: multicast	Needed by the Multicast framework in order to identify if a flow is multicast or not. The overlay multicast approach does not maintain the multicast address in the core part of the network.
A	Adaptation allowance	0: Adaptation cannot be applied 1: Adaptation can be applied	Needed by the network node (MANE) in order to know if it is allowed or not to apply its adaptation feature.
M	Service Management degree	0: Partially managed 1: Fully managed	Needed by the network node (MANE) in order to know the degree of guarantee associated to the passing flow: whether it needs to be threated with hard or loose level of QoS guarantees.
E	Reserved	To Be Determined	Reserved for future extensions.
STYPE	Service Type	1: LiveIPTV 2: VoD 3: Push Content 4: Audio/Video Call	Needed by the network node (MANE) in order to identify to which Media Service corresponds the passing flow and then apply accordingly the correct actions, previously enforced by the CAN Manager

		5: Audio/Video Conference 6: P2PTV 7: P2PVoD 8: WebTV 9: WebVoD 10: Video-Centric Interactive Service 11: SEM-Enhanced VoD Others: To Be Determined	according to the agreed CANP-SP SLA.
SST	Service SubType	Defined according to each STYPE by Service Provider, when necessary	Extension of the SSTYPE field to provide deeper granularity to the Service Type categorization.
SPR	Service Priority	0: No priority 1: Low priority 2: Medium priority 3: High priority	Used by the network node (MANE) in case of necessity to take strong measures such as dropping, queuing or delaying packets in order to meet the agreed SLAs. These actions will be performed according to the priority field. The authorisation and manner of insertion of such information by the SP will have to be specified on the agreed SLA. If applicable, MANE will consult the appropriate table and mark the corresponding flow with the indicated DSCP. This will be detailed in D6.2.
VCANID	VCAN identifier (available after the SLA is contracted)	Values that correspond to different VCANs	Needed by the network node (MANE) in order to consult the Policy Table, apply the indicated MPLS Label and forward the flow through the appropriate interface.
RES	Reserved Fields	To Be Determined	Reserved Fields for future usage.

Several objectives should be envisaged when selecting the position in the data packets to insert CATI (probably not all can be fully satisfied simultaneously):

1. Solution for insertion format should be as independent as possible of application/service/content-type:
    a. Fixed position or easily findable positions (by the MANE) of the CATI fields in the data packets;
    b. CATI field position should not be dependent on session specific parameters in the same service type class;
2. Compliancy with current TCP/IP stacks and standards as: IETF, ISO/IEC, etc.;
3. Open solution (if possible) w.r.t. future developments;
4. Moderate/low complexity of implementation in the HB/servers which should insert CATI in the packets.

The CATI insertion can be performed at different levels of the ISO/OSI model: network (e.g., IP header, transport (e.g., UDP header), application (e.g., RTP, FLUTE, etc.). It was also considered to insert CATI between L3 and L4 headers. This solution would actually define a new sub-layer 3.5 in which a new header (CATI) would be added between IP and

UDP/TCP headers Table 5. The method that has been considered as most appropriate in **ALICANTE** for in-band CATI insertion is through the exploitation of the **RTP Extension headers**. Analyzing the possible solutions, the advantages offered by the usage of RTP header extension seems to be important (given that a lot of media flows are transported on it and it is used as the main protocol in **ALICANTE** for Media Services):

- The Content servers/HB media servers only perform processing above the transport layer (no IP level processing);
- Relatively easy to be extracted by MANE;
- Natural position –- in the stack – for content/service-related data;
- Independent of L3 protocol (e.g., IPv4, IPv6).

Therefore, the first option for insertion of the CATI in the RTP extension header will be applied in the current implementation.

Table 5. Summary of characteristics of CATI insertion methods

Insertion Method **Characteristics**	**IPV4 header extension**	**RTP header extension**	**Packet** **Payload**	**New 3.5 layer**
CATI extraction by the MANE	Simple (fixed position)	Relatively easy	Complex (variable position)	Simple (fixed position)
Independency of other layers protocols	At MANE level: independent of L4 or higher layers	CP/SP: independent of L3 protocol (IPv4, IPv6)	CP/SP: independent of L3 protocol (IPv4, IPv6)	CP/SP should insert CATI at layer 3.5
Extendable for IPv6	Straightforward (Flow label or using additional headers)	Usable	Usable	Usable
Compliancy with current IETF stack	Yes	Yes	Yes	No
Independency of A/V coding formats	Yes	Yes	No	Yes
Fixed position (related to IP header)	Yes	No	No	No

The Real-time Transport Protocol RTP is the most common application protocol for end-to-end network transport of data with real-time characteristics, such as audio and video. RTP runs on top of UDP to make use of its multiplexing and checksum services. The RTP specification [127], describes the RTP header as having the fixed format presented in Figure 5.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5. RTP Header Syntax

If the X bit is set, the fixed header represented in Figure 5 is followed by one header extension, providing the capability to extend the RTP header. The extension mechanism allows for carrying additional information in the RTP data packet header. This mechanism is designed so that the header extension may be ignored by other interoperating implementations as depicted in Figure 6.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      defined by profile       |           length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        header extension                       |
|                             ....                              |
```

Figure 6. RTP Header Extension Syntax – 2B

The header extension contains a 16-bit ‘length’ field counting the number of 32-bit words in the extension, excluding the four-octet extension header (therefore, zero is a valid length). The RTP specification is characterized by the two following drawbacks concerning the insertion of the RTP header extension:

- At most, one extension per RTP packet is permitted by the existing header extension method;
- Moreover, the specification does not specify how the 16-bit header extension identifiers are allocated to avoid collisions.

The RFC5285 specification [128] removes the first drawback by defining a backward-compatible method to carry multiple header extensions in a single RTP packet. It also removes the second drawback by specifying that these extension elements are named by URIs; a Session Description Protocol (SDP) method is defined for mapping between the naming URIs and the identifier values carried in the RTP packets. The RTP header extension is formed as a sequence of extension elements, with possible padding. Each extension element has a local identifier (ID) and a length (len).

There are two variants of the extension: one-byte and two-byte headers. For **ALICANTE** aims, given the CATI structure, the one-byte header form is preferred, since the number of extensions in any given RTP session is just one and the extension itself is small (two 32-bits word). In the one-byte extension element variant, the 16-bit value labelled in Figure 7 as "defined by profile", takes the fixed bit pattern 0xBEDE and each extension element starts with a byte containing an ID and a length, as in Figure 8.

```
        0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|  ID   |  len  |
+-+-+-+-+-+-+-+-+
```

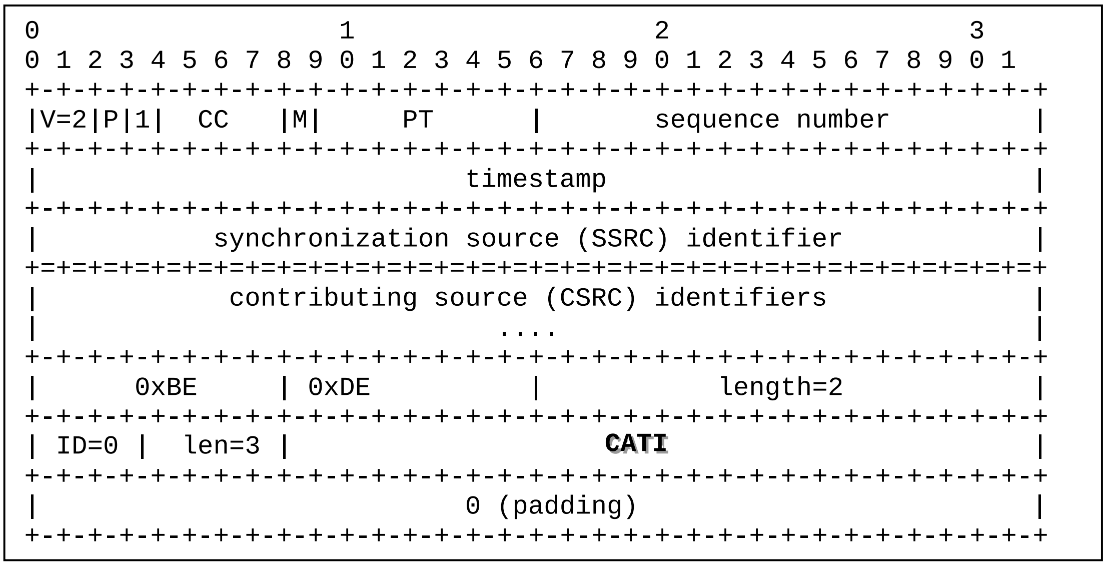
Figure 7. RTP Header Extension Syntax – 1B

The header extension is identified by the 4-bit ID in the range 1-14 inclusive. The 4-bit length is the number minus one of data bytes of the header extension element following the one-byte header. In the general case, a sequence of extension elements, possibly with padding, forms the header extension defined in the RTP specification. There may be as many extension elements as fit into the length, as indicated in the RTP header extension length. Since this length is signalled in full 32-bit words, padding bytes, having the value of 0, are used to pad to a 32-bit boundary.

Since the CATI has a structure formed by 24 bits, one single extension element composed by two 32-bits words is needed; the first byte is allocated for the header extension, the next 3 bytes are reserved for the CATI. Hence:

- The 16-bit value labelled in Figure 6 as "defined by profile" takes the fixed bit pattern 0xBEDE,
- The length field in Figure 6 is equal to 2,
- The ID field in the RTP header extension (Figure 7) is equal to 0, or any other integer value,
- The len field in the RTP header extension (Figure 7) is equal to 2 (a total of 3 bytes of the header extension element following the one-byte header, minus 1).

As a summary, an example of insertion of CATI in the overall RTP header is reported in Figure 8.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|1|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0xBE     | 0xDE          |           length=2            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ID=0 |  len=3 |                    CATI                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           0 (padding)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8. Example of CATI insertion in the RTP Header

4.4. Content-Aware Network Elements in ALICANTE

The main network element that we propose within this thesis is called Media-Aware Network Element (MANE). It has been conceived in direct conjunction with the overall ALICANTE architecture. The MANE is supposed to replace border routers at the edges of network domains and will be able to identify and classify incoming traffic. In order to overcome some of the performance and scalability issues that are inevitable due to the complexity of the classification mechanisms, the target is to propose a system that will be modular, allowing several levels of identification precision and the significant granularity during classification process. The configuration of the system will depend on the Network Provider's needs, in terms of traffic handling and QoS support. For example, a NP could select to classify only RTP flows, hence only activating the related options. In the next sections we will describe the basic functions under which the MANE is built along with its architecture.

4.4.1. Flow-Awareness

This section provides information on the design and implementation of flow-awareness functions inside MANE. A flow is determined as a set of packets related to an instance of some applications/services observed at a given point in the network with an inter-packet interval less than a few seconds. This definition is rather broad and in the context of flow identification, it can be narrowed down. The flow may, for instance, include several TCP connections used for transferring a web page or could comprise the RTSP and RTP streams of a streaming video service. The flows occur inside a network within sessions. A *session* is observed at any point in a network as a sequence of flows separated by idle periods, also known as think-times.

Incoming data packets that arrive at the ingress MANE are analyzed and classified according to the content carried in the packet payload. In order not to apply this intensive work on every single packet for new incoming flows, the first few packets of the flow identifies the content-type and this information are kept in memory along with the 5-tuple of protocol header information (i.e., PktCtx), as defined in next sections. This information, along with a unique hash key, calculated from each 5-tuple, defines the flow. For every other incoming packet that belongs to an existing flow, only the hash key needs to be calculated. This allows the handling of incoming traffic per flow and not per packet, increasing efficiency and minimizing possible delays. In order to prevent the flow table from overflowing, a separate mechanism monitors the flow activity (for UDP) or the protocol signaling (for TCP) and removes the terminated flows from the table. The flow-awareness enabling function is developed on Linux, exploiting basic open source software libraries (e.g.., LibPcap [107]).

The software modules realizing flow-awareness functionalities are implemented in C++. The flow-awareness functionality is applied at ingress MANEs using a memory resident hash table. After the flow has left the ingress MANE, it is being forwarded using MPLS, avoiding the need for re-identification of the flows in the core routers and thus resulting in a more scalable solution.

In this context, a special *Flow* class is implemented to keep track of the information for each new detected flow. The information used for identifying the flows is the IP addresses, port numbers and L4 protocol type. This information is accessed through the *LibPcap* Open Source Linux library [108] that is used to capture packet information as well as the payload and supplies the appropriate data structures. The actual data packet path through the data plane is not affected at all. Only a copy of the appropriate packet data portion is accessed by the user space respective module.

Although *LibPcap* implementation is rather stable and quite tested in various open source projects, it currently introduces performance limitations regarding the packet rate it can handle. Generally, packet capturing performance, based solely on software, is a demanding process mainly because the main effort of the network card drivers is to pass the data packets to the kernel and this limits the availble CPU cycles. Experimental results as presented in [109] and [110] depend on the OS used (BSD, Linux, Linux NAPI), on the software system calls implementation *(*i.e., *device polling, PF_RING, netfilter)* and on the hardware capabilities (CPU speed, TCP offloading, libpcap hardware acceleration). Considering the case of a PC with 1.8 GHz XEON CPU with 1 GB of RAM set to capture 1 Gbps traffic of 64byte packets, the calculated *captured packets* over *total packets* ratio is around 30% with a 100% CPU utilization. This ratio can even reach 100% if hardware acceleration is used. Furthermore, there are also limitations due to the PCI architecture and supported speeds, as well as the fact that most Gbit network cards and their low level OS drivers have not been designed for capturing thousands of packets per second in promiscuous mode.

In order to overcome these problems and provide a more efficient capturing mechanism, the flow management can be applied only on ingress routers using a simple system with a hash table. This will remove the need for keeping track of flows on all nodes of the core network and result on a more scalable system.

The *Flow* class was created for the purpose of holding the information for each new detected flow. The type of service of the flow is to be subsequently determined as well as some statistics such as the flow count and packet count for the flow. The *FlowHash* class has been created to keep the state of the detected flows. As its name implies, the implementation relies on a hashmap where the flows are stored. In the hash table, statistics for each flow are kept and updated. The information held inside the Flow Hash table is illustrated in the Table 6.

Table 6. Flow Hash Table of the software-based flow-awareness module.

Entry Counter		A counter for the number of the flows currently active
Hash Key		Key generated by the 5-tuple and used for fast indexing and searching
IP Address	**Source**	Layer 3 (IP) protocol headers
	Destination	
Port Number	**Source**	Layer 4 (UDP/TCP) protocol headers
	Destination	
Protocol Type		The Layer 4 protocol type (i.e., TCP, UDP)
Content-Type		Filled with the content-type after the Content-Awareness has taken place
Activity		Packets passed for this flow
Timer		A timer that monitors the idleness of the flow in order to keep or delete after expiration

4.4.2. MANE General Architecture

The main objective of this chapter is to present the design of the MANE based on a standard Linux kernel that encapsulates all the routing and forwarding capabilities available on a Linux system and, at the same time, exploits content-aware functionalities using specific developed software modules.

In order to manage and control the packet classification, queuing and scheduling processes needed to support traffic classification, QoS and MPLS, the developed software modules interface with the related user-space modules that natively control these processes. The result is that the content-aware modules are able to enforce certain policies and decisions based on the identification of the content type.

The architecture is divided into three distinct entities: (i) User Space; (ii) Kernel Space; (iii) Physical Data Plane. Only the key functions and modules needed for the implementation of MANE are included in each entity.

Design Advantages: the implementation is built around commercial of the shelf (COTS) hardware equipment easily upgradable and versatile; re-uses stable and well-tested GNU software modules and libraries; eliminates the need to develop Layer 3 and Layer 4 functionalities from scratch that are adequately integrated to this implementation; allows easy integration of new modules and functionalities.

Design Drawbacks: since the implementation is Software based and the hardware binding is high level, the performance of this implementation has limitations in reference to (i) porting to other architectures and (ii) performance limitation inferred by the OS architecture (i.e., system calls), etc.; problems/changes that affect the operating system kernel architecture might affect the performance/operation of the various User Space developed modules.

To support CA functionalities DPI techniques are utilized for classifying incoming flows to service types. The classification of content is performed in a per flow basis allowing for better results and requiring only a minimum set of packets to be examined since subsequent packets to a classified flow need not be processed further. This approach leads to the diminishing of required processing for the classification and improves greatly the overall scalability of the system. Details on the whole process can be found here [159].

In Figure 9 is presented a draft overview of the processing stages that an incoming packet has to pass. Incoming packets, at the ingress Label Edge Router (LER) are checked over installed filters (installed through the *iptables* utility at the kernel). The use of the filters is twofold, the first is to apply the appropriate traffic policy (if needed) and the other to mark the incoming packets using a marking scheme that is later used by the MPLS forwarding in order to accordingly assign the marked packet to the designated LSP and traffic class (exploited by DiffServ). In turn, the marked packets are passed to the MPLS Forwarding, which is the kernel module that forwards the packet through the appropriate physical interface, while applying the appropriate MPLS label. Finally, the packet is queued for transmission.

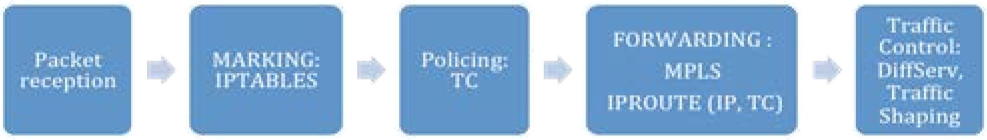

Packet reception
MARKING: IPTABLES
Policing: TC
FORWARDING : MPLS IPROUTE (IP, TC)
Traffic Control: DiffServ, Traffic Shaping

Figure 9. Packet path through the MPLS ingress LER

4.4.3. Functional Blocks and Interfaces

The architecture is based on a Linux distribution on x86 hardware. Along with the development of **ALICANTE** specific modules and software, this architecture integrates and exploits some of the already available Linux-based software modules and utilities. Most of the developments are introduced in the user space part of the architecture.

The User Space contains all the control and management modules that are currently available on the Linux OS integrated with those being developed to enable content-aware functionalities. To this extent, as shown in Figure 10, the modules that are re-used are:

- IPTABLES: a generic table structure for the definition of rule-sets [130]. Each rule within an IP table consists of a number of classifiers (iptables matches) and one connected action (iptables target);
- IPROUTE: a collection of utilities (suite) for controlling TCP/IP networking (IP utility) [131] and traffic control in Linux (TC utility) [132];

- MPLS Control Utility is a simple command line utility for creating and deleting MPLS ports, switch paths, ingress mappings, and egress mappings [133].
- LibPcap Library: a library used for packet capturing and header parsing, since it is a software implementation the performance has certain limitations [107].

The kernel space contains the modules that operate at kernel and are controlled by the utilities at User Space. The MPLS related kernel modules that affect the native routing and forwarding process, which are related to the QoS support, traffic shaping, are: MPLS Forwarding, MPLS LIB and MPLS Classifier.

- MPLS Forwarding module processes incoming packets and makes forwarding decisions based on the policies provided by the MPLS signalling protocols and the MPLS LIB;
- MPLS Label Information Base (LIB) is a table; populated either statically or dynamically (by label distribution protocols) that specifies how to forward a packet. This table associates each label with its corresponding Forwarding Equivalent Class (FEC) and the outbound port to forward the packet to. The LIB is typically established in addition to the routing table and Forwarding Information Base (FIB) that traditional routers maintain;
- MPLS Classifier is used for specifying Quality of Service (QoS) constraints on packets traversing an LSP. Recent versions of the Linux kernel support differentiated forwarding of packets based on a flexible architecture of filters, classes and queuing disciplines. In particular, it is possible to filter outgoing packets based on a special traffic classification index, which can be specified as part of an MPLS switch table entry (stored in the MPLS LIB). The index filter can be used to pass packets to different traffic classes based on the value of this index. Each class may be associated with a different queuing discipline supporting differing forwarding constraints (e.g., limiting transmission bandwidth).

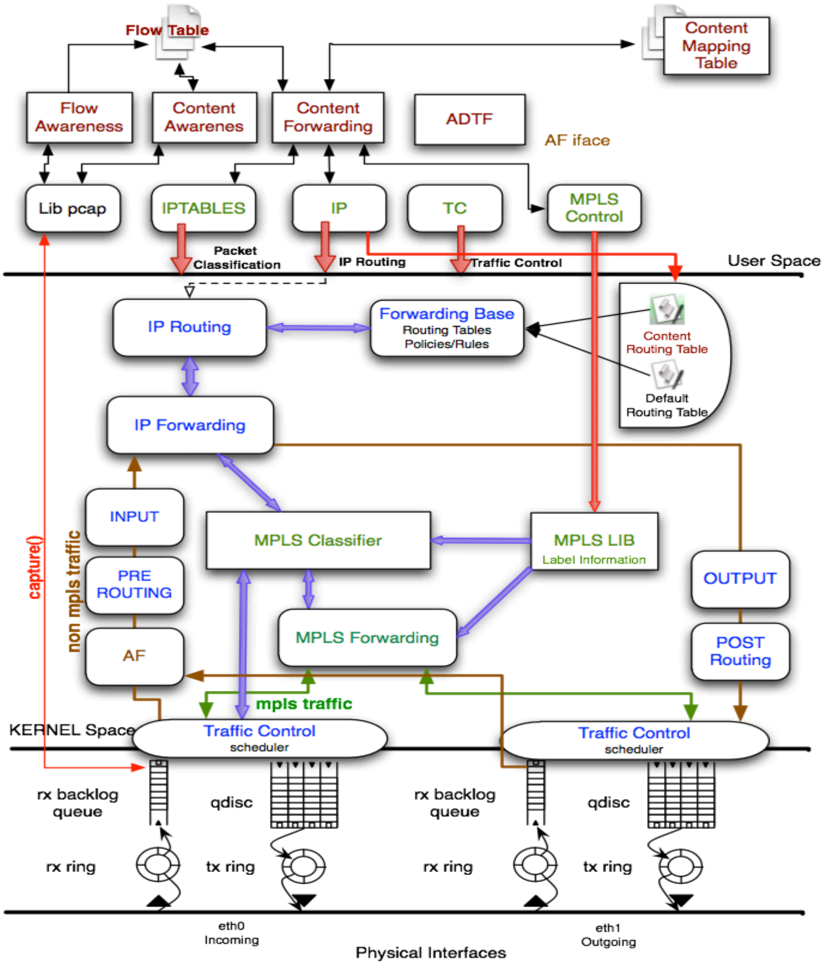


Figure 10. MANE architecture description

4.5. MANE Evaluation

This section presents the experimental results of the forwarding performance evaluation of the MANE. The objective is to evaluate the maximum non-drop rate (NDR) forwarding performance of MANE. NDR is also known as the maximum load without packet drops, beyond this point the MANE will start dropping packets. The NDR was measured against IPv4 traffic over Gigabit Ethernet.

The experimental scenario involves testing the MANE in a standalone setup as depicted in Figure 11. The MANE is placed between a Software Traffic Generator and Traffic Sink where traffic is collected for analysis. Additionally, MANE performance is also compared against the forwarding performance achieved by the same hardware without the content-awareness functionalities.

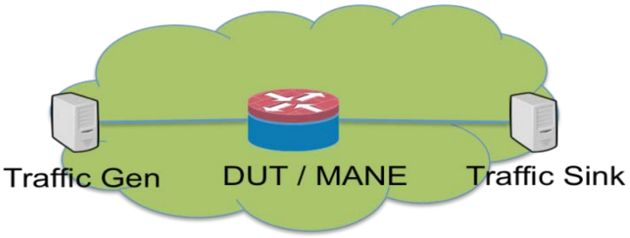


Figure 11. MANE related Experiment Setup

In the following experiments, the MANE is implemented on workstation hardware with Intel® XEON CPU E5645 @ 2.4GHz with 1Gbps PCI-X network cards. The testing tool was the D-ITG software based traffic generator [134].

The testing procedure is defined by RFC 2544 and is also adopted for the MANE performance evaluation. For both series of tests, the traffic generator generated traffic at 1Gbps for the following frame sizes: 128B, 256B, 512B, 768B, 1024B, 1256B and 1512B. The CRC Ethernet checksum is not included in these packet sizes. Each test was effective for 30sec and was repeated 3 times. The results are illustrated in Figure 13 and the detailed results in Table 7.

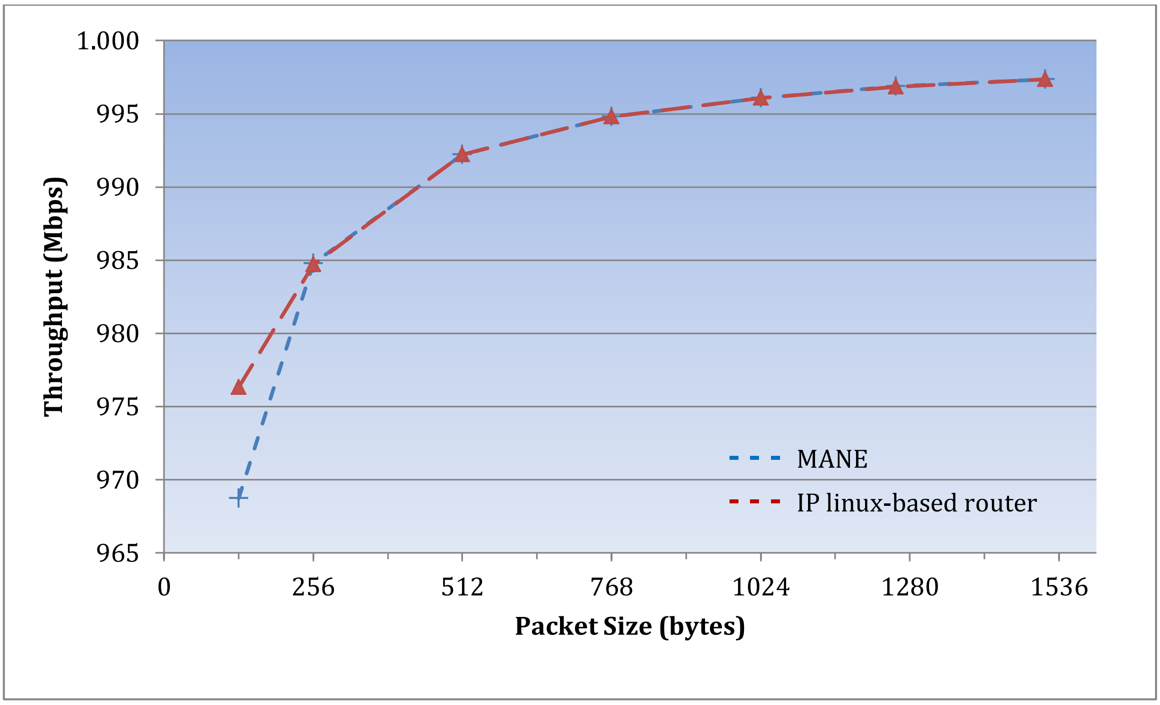


Figure 12. Packet Forwarding Performance of MANE

Table 7. Throughput of MANE and IP Router

Frame size [Bytes]	*MANE traffic [Mbps]*	*IP Router [Mbps]*
128	585	596
256	985	985
512	992	992
768	995	995
1024	996	996
1256	997	997
1512	997	997

As it can be easily deducted from the above, the NDR forwarding performance of the MANE is the same as the legacy IP router configuration (running on the same HW). This is anticipated by the fact that the enhanced MANE functionalities, although they affect the CPU utilization, do not affect the forwarding performance. MANE during the timeframe needed for a classification and identification, does not intercept the packet-forwarding path through the Linux kernel structures, but copies the needed portions of data from the packet. When the identification has achieved its goal, the flow is automatically classified, accepting the fact that some (few) initial packets will be forwarded using the default policies.

Since the bottleneck of the implemented MANE appears to be related to the LibPcap limitations [135], the following results present the capturing packet rate achieved by the MANE. From these results, it is apparent that the LibPcap performance is adversely affected by the existence of small packets up to 512 bytes at gigabit rates. However, as the packet’s length increases the performance, the MANE packet capturing function produces zero packet drops.

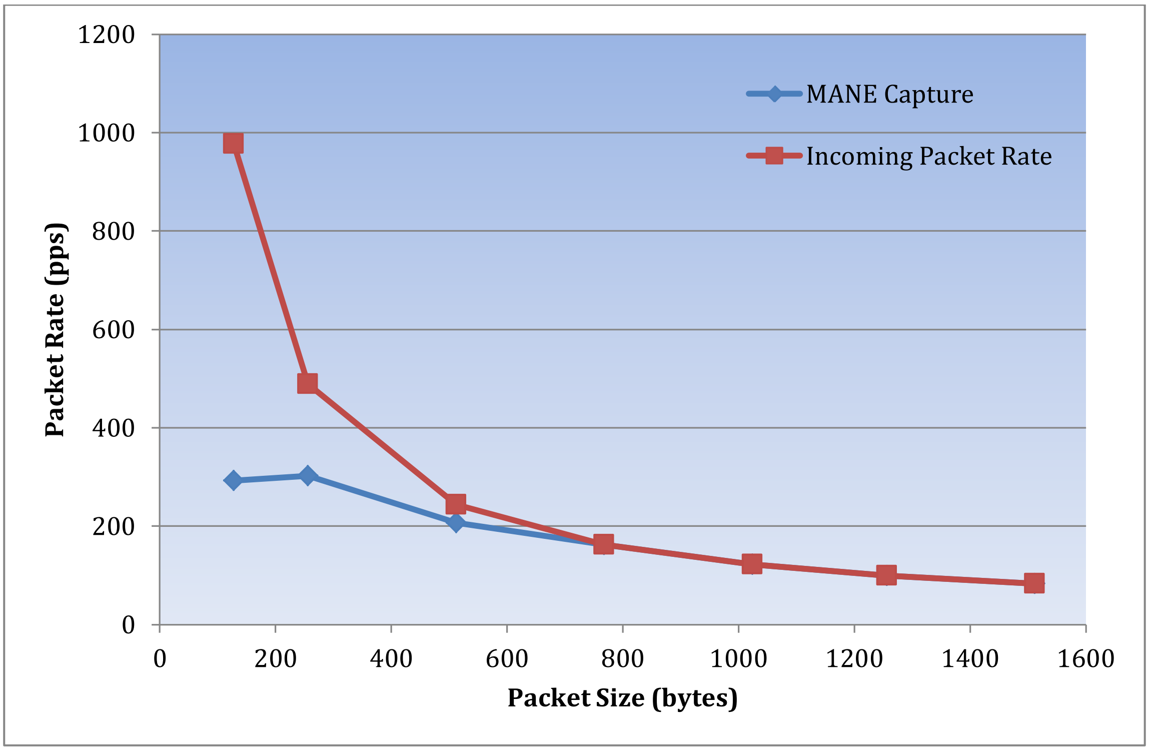


Figure 13. MANE Capturing Performance

In order to be able to also check the performance of MANE under different HW, an ordinary PC was used. The PC HW had the following characteristics: CPU Intel(R) Core (TM) 2 Quad CPU Q8400 @ 2.66GHz, with 4GB RAM and a dual port INTEL PCI-X network card. The results are illustrated in Figure 14. The experimental procedure followed once again the provisions of RFC 2544 and involved the following three MANE setups:

i. MANE: off, MPLS: off;

ii. MANE: on, MPLS: off;

iii. MANE: on, MPLS: on

The results showed that the performance is slightly affected by the downgraded HW capabilities, mostly for packets smaller than 500Bytes. This is the shifting point in all the cases tested. The same limitation for the LibPcap is true also for this setup. The MPLS performance is deteriorated for very small packets (i.e., 256 and beyond). This is due to the fact that the CPU load reaches 100% for those packet sizes, as Figure 14 illustrates. The results presented in Figure 15 showed very good performance of the MANE under load of a single flow at line rate.

The MANE implementation was evaluated by measuring the packet forwarding performance and the capabilities of the content-awareness framework against increasing number of incoming flows. It was presented that the forwarding throughput of the MANE and IP router is similar and both of them decrease for lower packet sizes (processing of small packets causes CPU overload and, as a consequence, losses). The operations at the ingress MANE related with MPLS (classifying, marking) are more complex and demand higher CPU utilization, resulting in decreasing the efficiency.

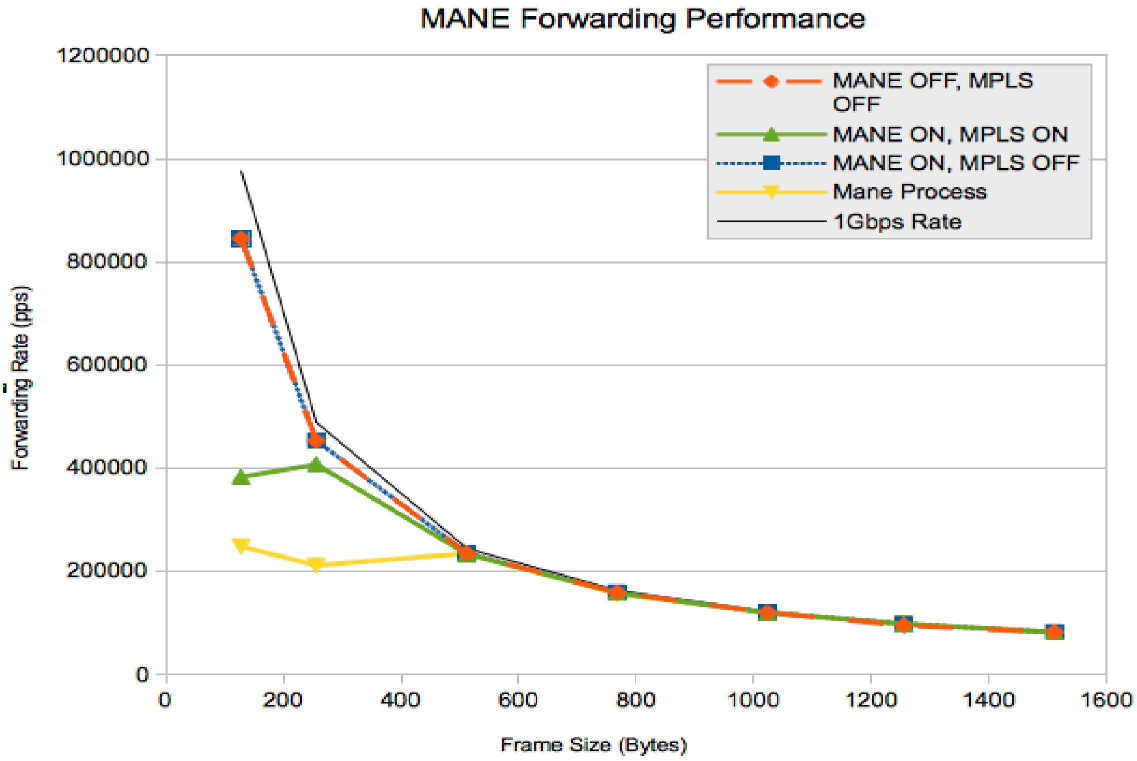


Figure 14. MANE Forwarding Performance

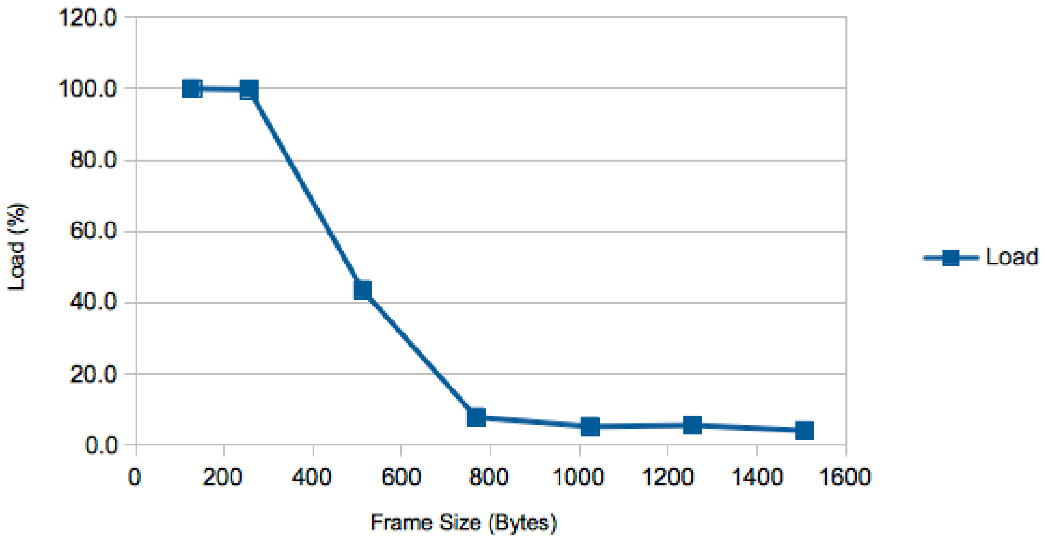


Figure 15. Aggregated CPU Load

4.5.1. Scalability of MANE operations

This section discusses about the scalability aspects of MANE. Although during design phase, major scalability issues have been considered, there are still some issues that require an additional study.

The packet inspection function required for the content awareness in the MANE raises important scalability concerns. The factors affecting the processing required for this function are:

1. Router Specific:
   a. Number of protocols examined
   b. Number of regular expressions used:
   c. Matching algorithms complexity;
2. Ingress Traffic Profile:
   a. Number of flows;
   b. Flow life-span, duration (short, long), with login flows being less expensive;
   c. Stateful protocol matching versus Static port matching

In order to estimate the scalability of the MANE content-awareness framework, we need to assess the correlation of the capability of capturing data packets at line rates with the new flow inter-arrival density. In this context, the device is tested under traffic load with variable flow inter-arrival density. In order to exclude problems caused by the network interface or limitation of its HW, the traffic is software generated and written in trace files for different values of flow densities and then is read by the content-awareness modules at the highest possible (allowed by the hardware) rate. The results produced by these experiments are illustrated in Figure 16. Four traffic traces are considered with new flow densities of 10, 100, 1000 and 10000 for 1400 bytes packet size.

As it can be observed, as the density of new flows per sec increases, MANE's capability to process the flows decreases. However, when the density of new flows reaches 10000 new flows per sec, then the MANE is able to process only half of them per second. The horizontal line illustrates the nominal Packet Rate required to reach at 1Gbps for 1400 bytes size packets. From this graph it can be deducted that 1000 new flows per second is handled easily by the MANE, for 10000 flows per second, the performance drops and the MANE cannot cope with the processing of so many flows. This limitation is easily explained by the fact that the MANE uses LibPCAP for the packet capturing. Finally, it can be deducted that the limit of MANE is between 3000-5000 new flows per sec. The plot of Packet Rate provided also from the figure, proves that after 6000-7000 new flows per sec, the MANE performance gradually reaches a threshold caused by the combination of HW and LibPCAP limitations.

.

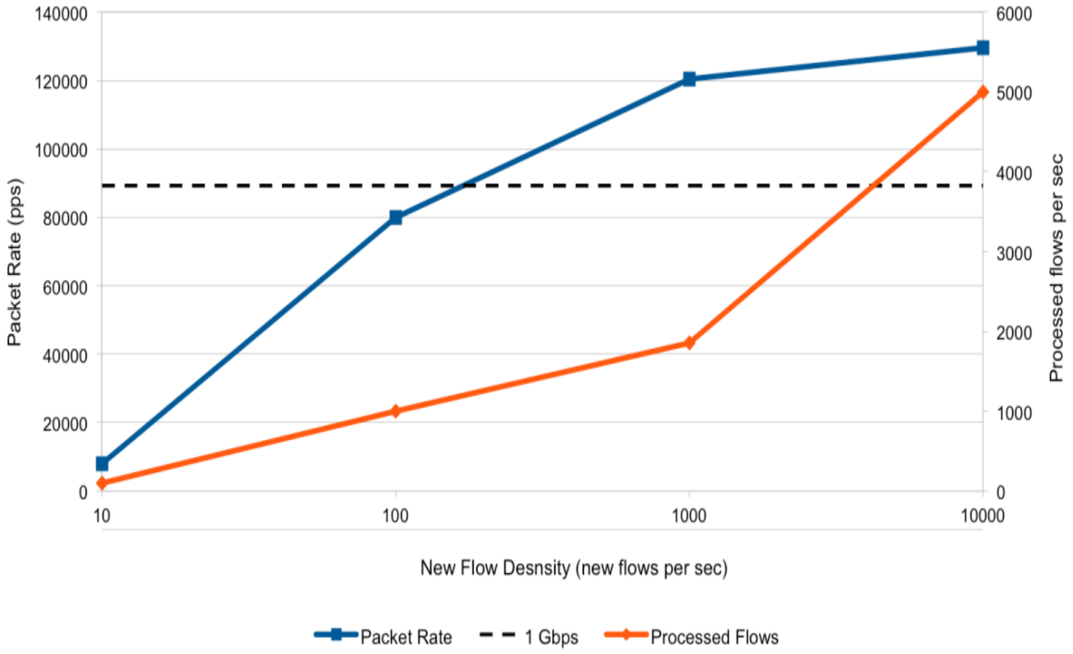


Figure 16. Packet capturing performance versus new flow density

4.6. Summary

This chapter presented a prototype Media-Aware Network Element (MANE) (i.e. an enhanced content-aware network module) that offers content type recognition and content-based routing/forwarding as a matter of guaranteed QoS/QoE provision in an end-to-end approach. The main objective of this chapter was to present the design of the MANE based on a standard Linux kernel that encapsulates all the routing and forwarding capabilities available on a Linux system and, at the same time, exploits content-aware functionalities using specific developed software modules. In order to manage and control the packet classification, queuing and scheduling processes needed to support traffic classification, QoS and MPLS, the developed software modules interface with the related user-space modules that natively control these processes. The result is that the content-aware modules are able to enforce certain policies and decisions based on the identification of the content type. Going a step further, we proposed a specific signalling mechanism: the Content Aware Transport Information (CATI) header that is injected from service generation nodes in selected fields inside application layer protocol headers with the target to simplify the content identification process. At last, the experimental results of the forwarding performance evaluation of the MANE were presented. They showed that the maximum non-drop rate (NDR) forwarding performance of MANE is the same as the legacy IP router configuration (running on the same HW) and reaches almost 1 Gbps. Regarding scalability, as the density of new flows per sec increases, MANE's capability to process the flows decreases with the limit to be 6000-7000 new flows per sec.

The work presented in this chapter has been reported in [157][159][160][161].

5. A Network Resource Management Framework In Virtual Content-Aware Infrastructures

5.1. Introduction

Using new content-aware network nodes (MANE), the network provider may define and configure a new virtual layer, containing one or more virtual content aware networks, customized for different purposes and offering services to upper layers. Therefore, the network provider becomes actually an enhanced network capabilities provider (i.e., CAN Provider) able to offer content-aware virtual network services. On the other side, the upper layer delivers content related information to the CAN layer and, thus, enabling a powerful cross-layer optimisation loop. The CAN layer may offer to the Service layers, a range of functionalities: starting from lowest level (i.e traditional IP based transport) up to the full CAN-related functionalities. The choice will depend on the routers capabilities, NP policy, and higher layer requirements.

For this reason, the creation of a hierarchical, autonomic, virtualisation-capable resource management system appears to be necessary in order to offer:

- Distributed management (each domain has its own management system) and orchestration of physical and virtual network resources;
- Efficient network resources provisioning utilising existing traffic engineering techniques and mature network technologies for QoS (MPLS, DiffServ), while embedding them in the Content awareness framework;
- Hierarchical monitoring and adaptation of network resources at network layer.

In this context, each autonomous domain (e.g. AS) will have at least one CAN Manager managing and controlling several virtual networks. The CAN Manager will be responsible for planning/configuring/maintaining the deployment of the virtual networks, based on the network provider's information delivered by the Intra-domain Network Resource Manager (Intra-NRM).

The Intra-NRM will be responsible for (a) monitoring the network layer resources, operation and status, (b) controling/configuring the resources inside the network domain like MANE elements, core routers, and (c) delivering information regarding the underlying network status and dimensioning to the CAN Manager, prior to creating, terminating and/or modifying a virtual CAN.

In the next sections we will present the general CAN and Network Layer architecture within ALICANTE, we will describe the main building blocks of the Intra-NRM's and explain their functionality. At the last part we will elaborate on the implementation of a validation environment that conforms to the design specifications, towards verifying the validity of the management network architecture.

5.2. The CAN & Network Subsystem

The CAN & Network Subsystem offers to the SP Virtual Content Aware Network services with different levels of QoS guarantees based on virtual networks in the Data Plane and constructed through Service Level Agreements (SLA) contracts concluded in the Management Plane. It also provides network information to the Home-Box so that they can enable the Network-Awareness feature Figure 17. The components permitting these features are:

- The Multi-domain CAN Manager: it instantiates the Virtual Content-Aware Network through multiple domains and multiple providers;
- The MANE (Media-Aware Network Element): it is the building block of the VCAN layer,which can be seen as the evolution of today’s edge routers, with advanced functionalities of Content-Awareness, including forwarding, monitoring and adaptation inside the network;
- The Intra-NRM (Intra Network Resource Manager): it is the component having the ultimate authority for configuring/controlling the network resources;
- The Core Network Nodes: they represent common network routers, enabled with Diffserv / MPLS functionalities.

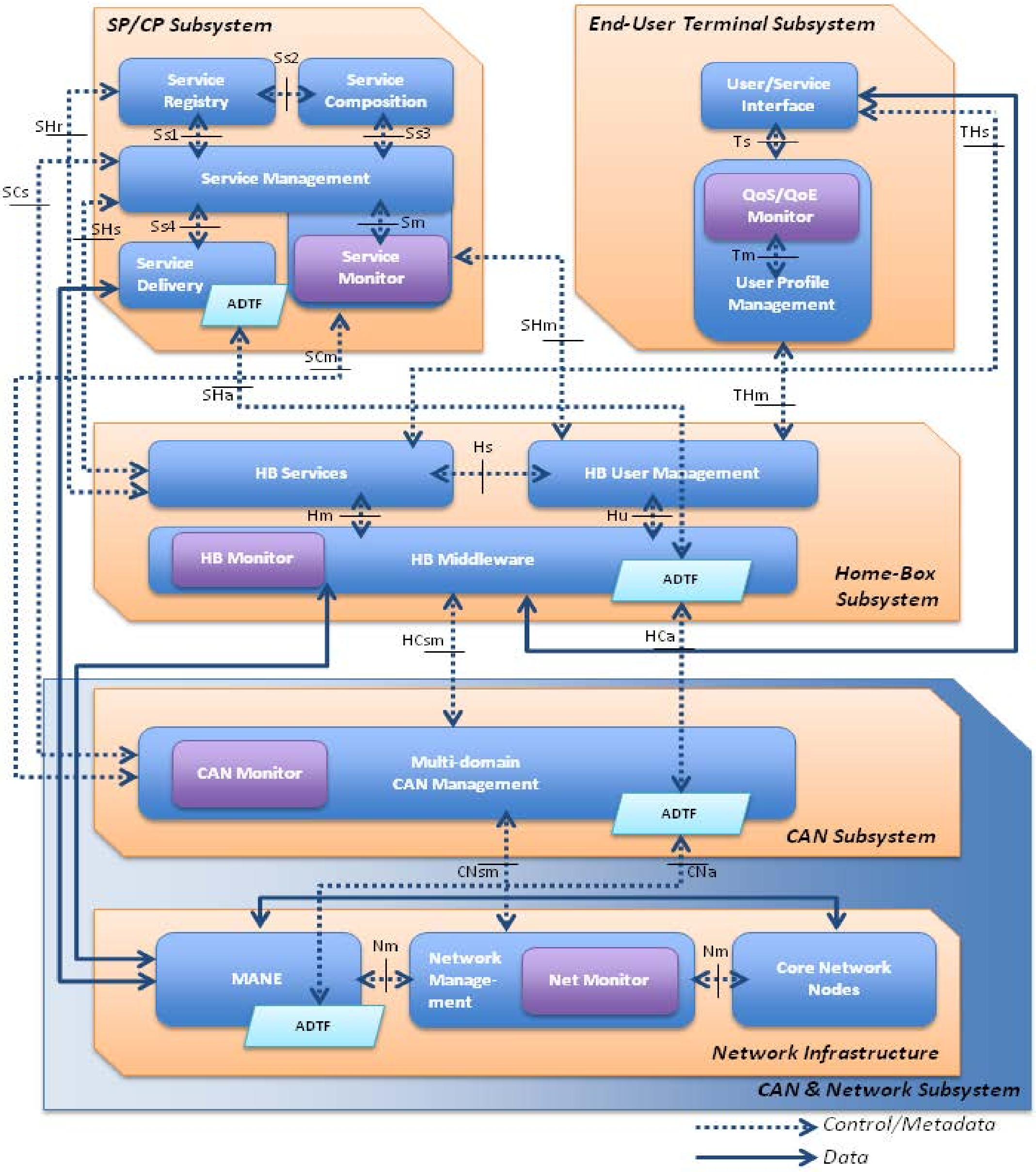


Figure 17. CAN and Network layers in ALICANTE

Focusing on the network infrastructure we detail the specific Intra-NRMs functions:

- Information delivery regarding the underlying network dimensioning and status to the CAN Mngr, prior to creating, terminating and/or modifying a VCAN. Between the IntraNRMs of adjacent domains, static interconnection agreements are supposed to exist in order to support multi-domain VCANs;
- Negotiation of VCANs with CAN Manager;
- Installation and operation of VCANs consisting of:

  - Control/configuration of the resources inside the network domain like MANE elements, core routers, etc. in order to support QoS enabled VCANs both for unicast and multicast communications;
  - Mapping of the VCAN logical pipes onto MPLS LSPs in case of unicast VCANs;
  - Control of the creation of the multicast trees inside the network domains and inter-domain;
- Configuration of the network level monitoring to audit the network layer resources, operation and status.

The following sub-sections will describe the key building blocks of the Intra-NRM and its interactions with the upper layers.

5.2.1. The Intra-domain Network Resource Manager Architecture

The Intra-NRM is the main entity to configure/control the network elements (MANE, core routers) of a single network domain. The MANE and the network layer (assisted by the CAN layer) may enhance and enrich service classification and differentiation, by extending not only to QoS provision according to content-type and associated policy information, but by adding other content-aware related attributes/processes to the corresponding services. Figure 18 shows the Intra-NRM functional architecture developed in [158], showing the interactions between functional blocks at a high level. The arrows depict the direction of the main flow of information between functional blocks, generally implying a certain configuration in the direction of the arrow. The architecture also shows the interactions between upper (CAN manager) and lower (MANE/core routers) network elements. The cross-layer monitoring on the right of the figure shows only the components directly involved with Intra-NRM.

The management plane functions are responsible for planning, dimensioning and configuring the control and data planes and interacting with CAN providers to establish virtual content-aware networks. While management plane functions are not as dynamic as control and data plane functions they are by no means static. Within Intra-NRM there is a continual background activity within the management plane at the epochs of the so-called resource provisioning cycles (RPCs). The intra-domain RPC which involves off-line intra-domain Traffic Engineering aims at proactively optimizing network resources to meet predicted demand and to build in sufficient spare capacity to avoid the burden of reconfiguring the network for each and every VCAN subscription or renegotiation, without the inefficiencies and costs associated with massively over provisioning resources.

In the next sections we will briefly describe the structure of the Intra-NRM's key building blocks and the interactions between them in order to justify their functionality.

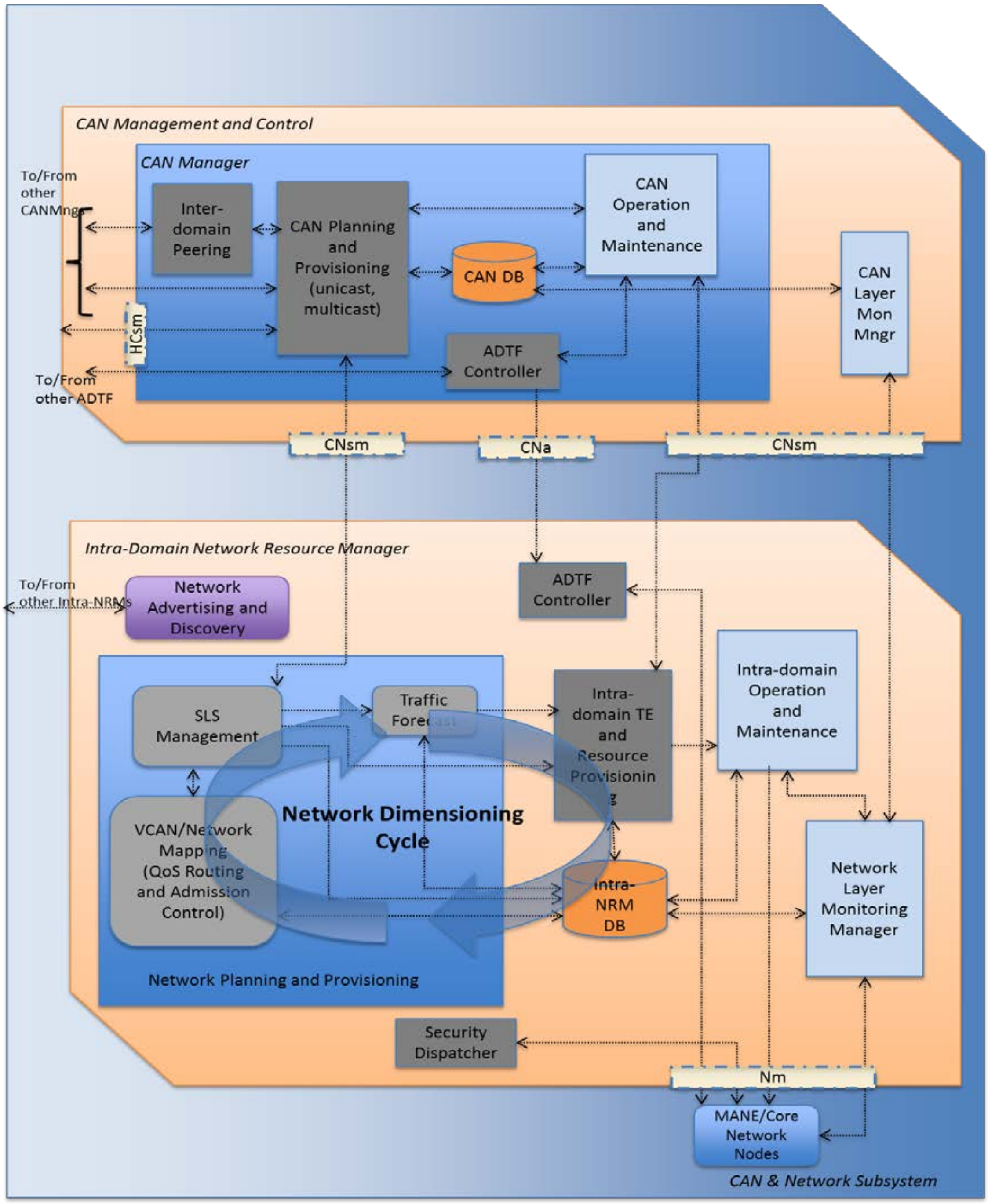

CAN Management and Control
CAN Manager
To/From other CANMngr
Inter-domain Peering
CAN Planning and Provisioning (unicast, multicast)
CAN DB
CAN Operation and Maintenance
CAN Layer Mon Mngr
HCsm
ADTF Controller
To/From other ADTF
CNsm
CNa
CNsm
Intra-Domain Network Resource Manager
To/From other Intra-NRMs
Network Advertising and Discovery
ADTF Controller
SLS Management
Traffic Forecast
Intra-domain TE and Resource Provisionin g
Intra-domain Operation and Maintenance
Network Dimensioning Cycle
VCAN/Network Mapping (QoS Routing and Admission Control)
Intra-NRM DB
Network Layer Monitoring Manager
Network Planning and Provisioning
Security Dispatcher
Nm
MANE/Core Network Nodes
CAN & Network Subsystem

Figure 18. IntraNRM Functional Architecture

5.2.1.1. Network Planning and Provisioning

Network Planning is defined as the off-line processes that are responsible for determining the type, quantity and geographical location of the physical resources required by a Network Provider in order to meet the predicted demand of its customers. According to the role of the Network Provider, the physical resources include, points of presence, routers and the communications links interconnecting them, as well as other equipment required for the operation of a network. On the other hand Network Provisioning is defined as the processes responsible for ensuring that the physical resources are deployed as planned and with the appropriate physical configuration. This is distinct from Traffic Engineering, which is responsible for managing the distribution of traffic, optimising the use of the deployed physical resources and ensuring QoS in a cost effective manner.

Traffic Forecast Block

The main objectives of the traffic forecast (TF) block are:

- To forecast traffic demand, based on a) existing and anticipated subscriptions/SLSes, b) historical data related to network usage and c) business policies. It is an important input to Traffic Engineering (TE) functions in order to dimension the network in terms of intra-domain network resources.
- To periodically check the validity of the results regarding the forecasted traffic demand and revisit, if necessary, the decisions taken related to TE.

The main result of the TF block is the estimation of an *intra-domain Traffic Matrix* (iTM), that will present the forecasted traffic demand between the AS network ingress and egress interfaces (MANE routers). Traffic demand is expressed in bandwidth units and for scalability reasons it shall be aggregated.

The TF block of the Intra-NRM communicates through the *CAN Planning and Provisioning manager@CANMgr*. In such a way, it is possible to:

- Monitor the established service agreements during each Network Dimensioning Cycle (NDC);
- Estimate the aggregated future demand.

The Traffic Forecast block provides to the *Network Planning* and *Provisioning (NPP)* block and in turn to the *intra-domain Traffic Engineering and Resource Provisioning (iTE&RP)* block the respective iTM in order to ensure that the local domain resources will be planned and engineered so that not only the established SLSes but also the anticipated to be ordered during the current provisioning cycle will be effectively accommodated.

Traffic Forecast detailed aspects and algorithms implementation are outside the scope of current investigation, given that such kind of blocks have been investigated in bibliography [139].

SLS Management

The main role of the SLS Management (SLSMngt) block is to negotiate with CANMngr the VCAN characteristics, while considering the CANMngr requests and Intra-NRM resources availability. The details of the negotiation protocol have been included in [137]. The template of the SLS between SP-CANP has been described in [138]. The content of the SLS for CANMngr to Intra-NRM negotiation is similar to SP-CANP one, except that the quantitative requirements are adjusted for this domain by the CANMngr.

The summary description of interactions between the CANMngr and Intra-NRM with respect of SLS negotiations is given in the text below.

Assumptions:

(1) Intra-NRM Data Base (Intra-NRM DB) contains among others:

a. Current network (i.e., domain configuration and resource) status (graph, nodes, links , capacities) expressed in the form of Resource Availability Matrix (RAM);

   b. Policies of Intra-NRM to decide the percentage of the total capacity that can be allowed for some classes of VCANs, depending on operator business objectives and current network status.

(2) CANMngr has summary information about this domain resources, (delivered by Intra-NRM), in order to split (if it is the case) a multi-domain VCAN in several parts, where each part of the VCAN is associated to a given network domain.

(3) SLSMngt at Intra-NRM contains the server part of the negotiation protocol. It finally accepts the SLS request or not (or maybe several negotiation steps can follow).

If the above assumptions are valid, then the interactions are:

1. CAN Planning & Provisioning (CAN P&P) delivers SLS requests for this network domain; the SLS includes the iTM and the QoS constraints. It negotiates the contract with SLSMgmt @Intra-NRM;
2. In order to check if it can accept or not the contract the VCAN/Net mapping block applies a combined algorithm both for unicast and multicast VCANs:
   a. Finding QoS enabled routes for all pipes within the iTM (basic metric or advanced metrics can be used);
   b. Admission Control based on the QoS class requested and level of guarantees needed;
   c. Logical resource reservation.

Note: if the Network P&P cannot fully satisfy the requested by the CANMngr contract, then it returns information on the partially satisfied request. It is for CANMngr to decide if:

- Partial fulfillment will be accepted or
- Re-negotiation contract process will start, given that the negotiation protocol is capable to support multiple steps negotiations.

3. In the successful admission control case (i.e., after deciding to accept the SLS) the SLSMngt block performs the following actions:
   a. Responds positively to CAN Manager;
   b. Stores the new VCAN contract data in Intra-NRM DB;
   c. Informs the TE&RP block about the new contract (indicates the place of appropriate data in DB);
   d. Informs the TF block about the new contract.

Note that the non-successful cases of negotiations will also be stored in the DB, in order to feed the TF with real data for the next NDC.

4. TE&RP block is responsible for mapping the requests onto LSPs and prepares the configuration of the multicast tree (in case that the VCAN is multicast);
5. TF will use not only past network status related information but also current status of the established VCANs in order to provision future network mapping and dimensioning. TF will produce an updated TM. It can be performed at the request of the SLSMgmt block if the number of SLS rejection increases.

Continuing the above (not directly related to SLS negotiation but as a consequence), the following actions are performed:

a. CAN P&P block at CAN Manager requests the VCAN installation in the network. This command is addressed to its O&M@CANMngr module;

b. O&M@CANMngr executes this command by requesting to TE&RP@Intra-NRM to install the VCAN;

c. O&M@Intra-NRM is instructed to install VCAN by the TE&RP @Intra-NRM. Here the real mapping onto LSPs is performed by using constrained routing paths delivered by the mapping protocol in the phase 2 of the above sequence;

d. Configuration Commands are sent to MANE and core routers by the O&M@Intra-NRM;

e. NetMon@Intra-NRM is instructed to monitor the network related characteristics of the VCAN.

Network Dimensioning Cycle

Depending on Intra-NRM's policy, the network dimensioning can be changed periodically, in terms of nodes, link capacities, bandwidth allocation for different classes of services, etc. This is called *Network Dimensioning Cycle (NDC)* and is represented by the loop in Figure 18, where:

- The TE&RP block receives TF information (TM) based on history of accepted SLS requests and forecasting information; it can apply an optimization algorithm to change network nodes and links re-dimensioning;
- At its turn, the TE&RP via Intra-NRM DB delivers to SLS Mgmt block, information about network resources (RAM).

An NDC similar solution has been already applied in other systems as those developed in [139]. The novelty in **our case** is that the NDC is applied by considering the virtual VCAN planes among which the real network resources should be allocated. The NDC is not mandatory to be done in automatic way; it is an offline TE process dedicated to optimise the network resource usage. Therefore, it can be triggered in one of several situations:

- When the percentage number of rejected SLS contracts is higher than a threshold established by the policies of Intra-NRM;
- Periodically (days, weeks, months - also depending on policy of the Intra-NRM);
- When upgrading the network (in terms of hardware or new network services offered);
- When the real usage of network resources is not equally balanced among different VCANs.

VCAN to Network Mapping Procedures

This section is focused on intra-domain planning and VCAN mapping, with QoS guarantees, thus continuing previous design for inter-domain part presented in [137]. A combined algorithm and protocol is proposed to perform jointly *QoS routing, admission control, VCAN mapping and resource reservation.*

The SP VCAN requests are expressed in the SLS. The needed topology is represented in an abstract way by sets of virtual links (called *Traffic Trunks*), belonging to a given QoS class. The SP knows the edge points of this VCAN, i.e., the MANEs IDs where different sets of Home-Boxes will be connected. This topology is treated by the CAN layer at two hierarchical levels: *inter-domain and intra-domain.*
The VCAN initiator CANMngr :

- Determines all Core Network Domains (CNDs) involved in a given VCAN;
- Splits the SLS parameters thus preparing request for individual network domains;
- Negotiates with all other CAN Managers to agree and reserve resources for the VCAN. A multiple domain VCAN belongs to some QoS class and therefore inter-domain QoS aware routing information is necessary in order to increase the chances of successful SLS establishment, between CANMngrs.

Note that the VCAN initiator CAN Manager should know the inter-domain *Overlay Network Topologies (inter-ONT)* and inter-domain available link capacities, in order to map VCANs onto network resources. The VCAN mapping is done on two hierarchical levels: *inter-domain and intra-domain.*

For *inter-domain* case, the initiator CANMngr determines the domains participating at VCAN. It knows the inter-domain graph (where each domain is abstracted as a node), inter-domain available link capacities (first the simplified case can be considered, i.e. only available bandwidth, but in general several other QoS parameters can be considered, e.g. delay). The SP request expressed in the SLS in abstract way as a TM plus other quantitative QoS requirements will be finally supported by real network resources (e.g. MPLS and DiffServ).

5.2.1.2. Intra-domain Traffic Engineering and Resource Provisioning

The purpose of intra-domain Traffic Engineering is to configure the intra-domain network in such a way that it will satisfy the requirements delegated from traffic forecast. The forecast provides the Intra-domain TE&RP block with demands for ingress, egress pairs and QoS constraints. Then, this block is responsible for the distribution of this traffic among the available network resources as efficiently as possible while honoring the given QoS constraints.

In ALICANTE, explicit LSPs can be used to configure different logical networks (VCANS) on top of the physical network. These LSPs can be thought of as virtual trunks that carry flow aggregates generated by the classification of the packets arriving at the ingress routers of the network into FECs. Therefore, a logical network composed by a set of explicit LSPs will be configured in the network to support the traffic flows of each Forwarding Equivalent Class (FEC).
The decision on what label to choose at the ingress LER can be based on either the fields in the packet headers and/or on a predetermined policy and/or current-state information. It will be the result of the communication between the CANMngr and the Service Provider and it will be based on a respective SLA.

A basic method for configuring labels is to aggregate different flows into *trunks*. A *trunk* is an aggregate of traffic flows that belongs to the same class of service, which means that all packets flowing in a trunk have the same MPLS header, including the 3-bit class of service field (also called *EXP* field). Different trunks can be routed along the same LSP. The only thing that distinguishes flows in different trunks is the class of service field. The ability to separate flows into different trunks naturally leads to better QoS, as it alleviates the interference among competing flows. The request for the construction of the traffic trunks is the result of the network planning and provisioning process.

MPLS has the capability to combine with DiffServ in order to enhance the QoS support inside the MPLS core network. More specifically, there are two approaches of how LER can attach a label to packet in order to keep the predefined priorities: (i) the EXP-Inferred approach and (ii) the Label-Inferred approach. Figure 8 shows these two approaches label distribution [133].

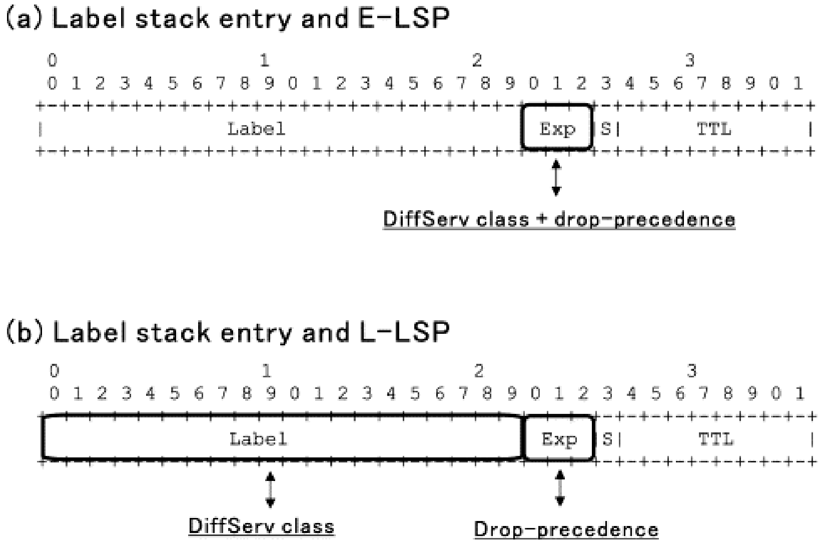


Figure 19. MPLS QoS-based Label distribution

- **E-LSP**: in this mode, the queue and the drop priority is inferred by the EXP field. The maximum supported classes are 8 (like IP TOS);
- **L-LSP**: in this mode the queue is inferred exclusively from label (like in IP/ATM multi VC) while the drop priority is inferred from EXP field. This combination will allow the support of up to 64 classes (like DiffServ).

For extending granularity, MPLS will be coupled with DiffServ in order to extend the QoS support and provide a greater granularity among the various QoS levels offered by the final architecture. In order to exploit DiffServ technology, a mapping rule is required between the *EXP* field of the MPLS header and the *DiffServ Code Point (DSCP)* field of the IP header. This rule will be exploited by the MANE ingress router in order to apply the appropriate marking to the incoming packets. Through the appropriate marking, which corresponds to a certain QoS traffic class and a specific LSP, the appropriate per-hop behavior (PHB) and forwarding is decided at each LER or transit router.

The decision on which flow to assign each mapping is a two-steps process:

1. It is the responsibility of the CANMngr which, after concluding the SLA agreement, forwards policing and QoS classification information for each contracted SLA to the Intra-NRM and consequently the latter to each MANE;
2. A Content Mapping Table predefines a mapping of content-types to QoS classes, for generic CA features operating upon services that do not contain CATI.

In this context, there are two possibilities when constructing the mapping table between the MPLS *EXP* field and the IP *DSCP* field yielding two different types of LSPs and consequently VCANs. In the first case, a single LSP holds different DiffServ Classes as illustrated in Table 8. In order to achieve this, a single MPLS *label* stack is used for all the MPLS routers in the same LSP, thus reserving the same PHBs.

Table 8. Multiple traffic class to single LSP mapping.

LSP id	**Traffic classes per LSP**
LSP1	EF, AFxx, CSx, BE
LSP2	EF,CSx
…	**…**

In the second case, each single LSP is mapped to one and only one traffic class, as illustrated in Table 9. In this approach, a unique MPLS *label* is attached for a specific EXP field value, which is mapped to a unique DSCP value.

Table 9. Single QoS class per LSP.

LSP id	**Traffic classes per LSP**
LSP1	EF
LSP2	AF11
…	**…**

In the realization proposed here**,** the DiffServ class scheme will support EF, AF1x (i.e., AF11, AF12), AF2x (i.e., AF21, AF22) and BE (Best Effort). An example can be found on the Table 10 considering an interface with nominal rate of 100Mbs. The last row describes the measures that the network will take for the excess information. It is possible to drop traffic or demote it (remark it and give it lower priority).

Table 10. Classes of Services

Class	**DSCP (HEX)**	**TOS (HEX)**	**CIR (Mbps)**	**EIR (Mbps)**	**Policy**
EF	0x2e	0xb8	40	40	DROP
AF1			20	60	DROP
AF11	0x0a	0x28	50%	50%	DEMOTE

AF12	0x0c	0x30	50%	50%	DROP
AF2			20	60	DROP
AF21	0x12	0x48	70%	70%	DROP
AF22	0x14	0x50	30%	30%	DROP
BE	0x0	0x0	20	60	DROP

As required for Assured Forwarding, we need a queuing discipline to support multiple drop priorities. This can be accomplished with GRED qdisc and take advantage of its probabilistic dropping mechanism. For the configuration of the respected values, the following formulas are used:

- **Maximum Threshold**=(Bandwidth Share * Desired Latency * Network Bandwidth) / (8 bits/byte* 1000 ms/sec):
- **Minimum Threshold**=1/2 * Maximum Threshold :
- **Avpkt**=Average Packet Length :
- **Burst**=(2 * MinThreshold + MaxThreshold) / (3 * Avpkt) :
- **Limit**=4 * MaxThreshold.

An example of the results can be found at the Table 11.

Table 11. AF Classes of services

GRED Virtual Queue	BW Share(ratio)	Latency-max (ms)	Average Packet Size (bytes)	Drop Probability (%)
AF11	0.5	800	1280	1
AF12	0.5	800	1280	2
AF21	0.7	400	1470	2
AF22	0.3	400	1470	4

5.2.1.3. Network Layer Monitoring

The purpose of the Network Monitoring system has been twofold: to provide CAN status data for facilitating CAN management and Adaptation decisions and also to provide Network Awareness to the service layer in the form of the so-called “Network Distance”. In order to serve these purposes, the architecture depicted in Figure 20 has been adopted. The figure also includes modules outside the Network and CAN environment, which also exploit CAN monitoring information [152].

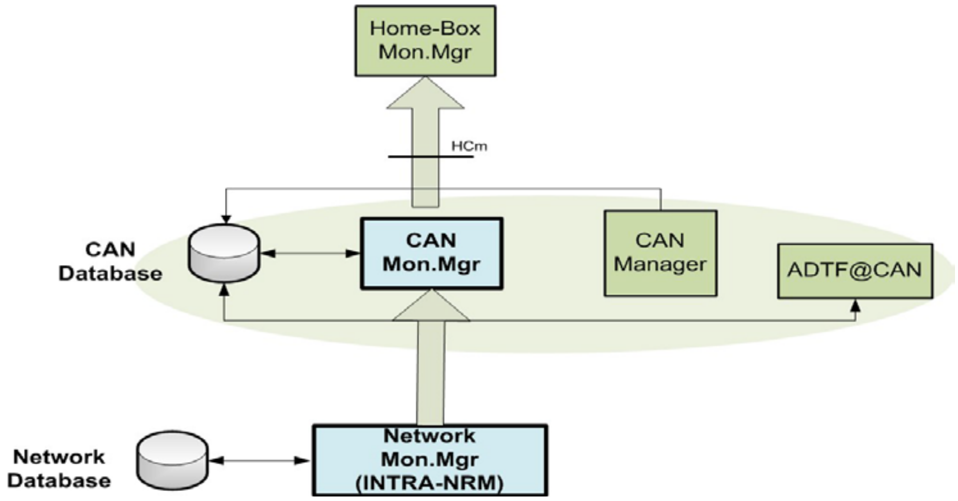


Figure 20. Monitoring at CAN and Network layers and associated modules

Monitoring at Network and CAN layers is undertaken by the Network Monitoring Manager (monitoring component of the Intra-NRM), which retrieves monitoring information directly from the network and the CAN Monitoring Manager, which maps the network monitoring information to the established VCANs and communicates the data to higher-layer entities (such as the HB and Service monitors).

The Network Monitoring Manager is a module continuously running and uses the Network Database to discover network and VPath topology, and periodically queries underlying MANEs via SNMP in order to retrieve passive and active monitoring information. These monitoring metrics are stored in the Network Database. The Network Monitoring Manager communicates Network parameters to the CAN Monitoring Manager.

The CAN MonMgr is a software module, which is co-located with the CAN Manager. Since monitoring is considered as a part of the overall management procedure, the co-location of these two modules under a single hardware platform responsible for the CAN environment is justified. The CAN MonMgr interfaces with: the CAN Database, to store both network- and CAN-level monitored parameters and also access VCAN information; the Network Monitoring Manager (at the Intra-NRM) to retrieve network-level monitored data; the HB Monitoring Manager, to expose Network Distance information for peer selection.

It must be noted that information on VCAN and SLA status, needed by external modules such as the Service layer monitors and the Adaptation Decision taking engines is made available via direct access to the CAN database. As shown in the figure above, the CAN

Monitoring Manager uses the CAN database for data storage and VCAN information retrieval. This database contains dedicated tables in order to act as a monitoring cache, containing recently derived and calculated metrics regarding the status of provisioned VCANs and VCAN pipes.

The CAN Monitoring Manager follows a two-threaded architecture. The first thread undertakes the processing of network-level measurements from the lower (Network) layer. In specific, it involves the following:

1. Periodically retrieves information on every one of the provisioned VCANs and VCAN pipes from the CAN Database;
2. Polls the Intra-NRM to retrieve passive measurements associated with the specific VCAN. The periodicity of Intra-NRM polling is adjustable;
3. Polls the Intra-NRM to retrieve active measurements associated with each pair of MANEs involved in the VCAN;
4. Calculates and derives VCAN-related metrics from the network measurements and stores them in CAN DB.

The second thread handles the response to external Network Distance requests originating from the HB layer. It performs the following actions:

1. Receives Network Distance requests from HB Monitoring Managers;
2. Maps HB-level data to VCAN-level data;
3. Retrieves most current data from the CAN DB;
4. Assembles and dispatches the response back to the HBMM.

A more detailed description of the requirements and architecture of the CAN/Network Monitoring Subsystem, including a complete reference to the monitoring metrics, are to be found in [140].

5.2.2. Validation Environment

Following the system's architecture, presented previously, this section elaborates on the implementation of a validation environment that conforms to the design specifications, towards verifying the validity of the management network architecture based on content-aware and virtualization concepts, via a series of preliminary experiments. In this context, it presents the implementation of a management framework as a process for the intra-domain network configuration that enables service classification via the virtual content aware networks, respecting several constrains and guaranteeing QoS related requirements. Towards addressing such challenges, various mechanisms are anticipated to function in the major managerial networking unit, namely the Intra-NRM. This unit operates by optimally assigning network resources to virtual networks, based on requests from upper layers (e.g.,CAN, Service layer). In this context, a number of preliminary experiments were designed and conducted under controlled conditions, elaborating on the overall system performance. More specifically, we focused on the configuration of the network using MPLS and DiffServ technologies. Analysis of the experimental results, verified the validity of the proposed architecture, establishing it as a novel solution with content-aware networks usage.

The testing environment is depicted in Figure 21. The basic building block of the core autonomous system is the Media-Aware Network Equipment (MANE). This network equipment enables the content-awareness at the core network level, as seen Section 4. At the same time, the MANE is acting as an MPLS router, in order to accordingly map the incoming content to the proper Class of Service and assign the appropriate MPLS header, in order the respective stream (that carries the content) to be routed in an efficient manner through the assorted Label Switched Path (LSP). In this way, the specific content flow is forwarded with a guaranteed level of QoS, implicitly deducted by the content.

The management system will react in a way that the End-User will get the requested service at a certain QoE level. The content-aware network elements that are between the End-User and the Content Provider will act accordingly in order to adapt the content to match to the End-User requirements and deliver the service at a certain QoS level. For each End-User, a context related profile is created that is kept at the Home-Box and is available upon request. This profile also contains information on whether the End-User has access to certain content and the priority that he might have among other End-Users. The End-User communicates with the Service Registry and requests a specific service. The requested service is matched to user’s contextual information profile and current status. In order for the service to be delivered in line with the previous requirements, the underlying virtual CAN layer is responsible for adapting the requested content and routing/handling the flow accordingly, in order to guarantee a certain QoS/QoE.

The architecture shown in Figure 21 presents the final demonstrator of ALICANTE project in Bordeaux pilot and is composed of:

- Full Service Environment: Service Registry, Service Manager, Content Servers, deployed on one Access network;
- Core network (single domain): MANE, Intra-NRM, CANMngr. All entities run as virtual machines using VMWare ESXi hypervisor;
- Full User Environment: User terminals and Home-Boxes with their associated software modules (QoE, User Profile, HB middleware/service layer).

In order to test and demonstrate the functionality of Intra-NRM on the platform, separate communication modules have been additionally implemented, in order to send some predefined provisioning information to the MANE/core routers. The implementation was partly based on existing components and open source code available from the Internet. Therefore, it is also pointed out that existing code has been adapted, updated, or modified and many parts have been newly implemented.

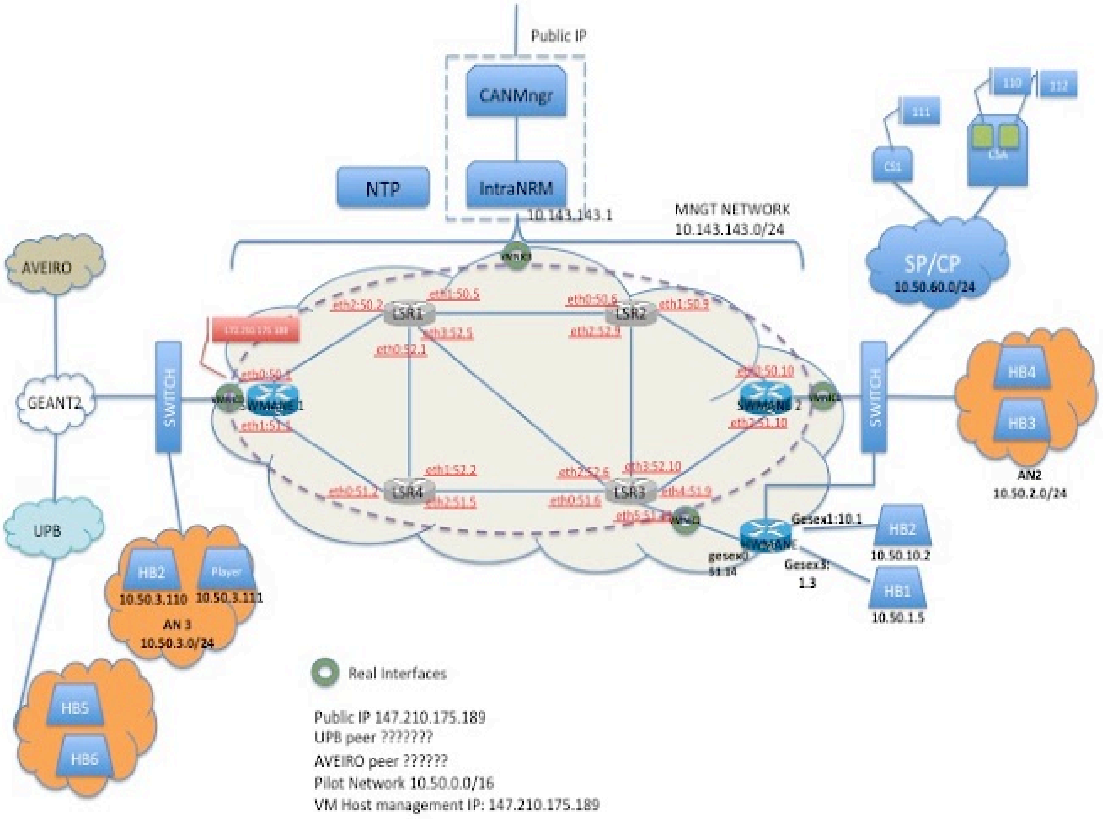


Figure 21. Testing Environment (Alicante Pilot topology in Bordeaux)

5.2.2.1. Validation Results

As mentioned in previous sections the MANE will be able to implicitly deduct the QoS requirements of different flows based on the flows content. For every recognized flow type, an appropriate instance of a virtual CAN will be assigned depending on the level of requested QoS guarantees. Efficient resource allocation and/or load balancing can be done in the network depending on traffic types and QoS requirements, by taking benefit from content awareness of MANE and based on operator policies, in terms of resource allocation. The CAN level will interact with the domain network resource management in order to perform mapping onto different QoS aware technologies (MPLS/Diffserv). Dynamic re-allocation (not frequently, in order to prevent instability) of the network resources between different CANs can be done, to assure the flexibility and efficiency of resource usage. In this context we will show below the network configurations that are necessary to be processed by the Intra-NRM in order to instantiate the differentiation in the flow treatment.

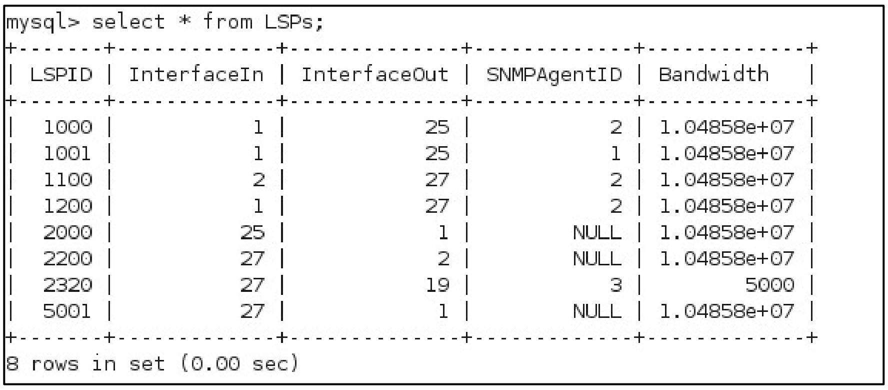
The Intra-NRM platform is launched with a shell script. The Intra-NRM's main controller automatically enables policy enforcement by establishing socket connections to MANE/core routers, loads notification information from the network database (netDB) and sends a request message for the installation of a configuration request state that is received from a notification file via the CAN manager.

The purpose of these tests is to validate the operation of network configuration in respect of:

1. MPLS functionality (LSP installation)
2. DiffServ functionality (support of the respective DiffServ classes)
3. Support of E-LSPs

The Intra-NRM is responsible to co-ordinate these functionalities in order to set up the network for the VCAN establishment. The steps below are followed:

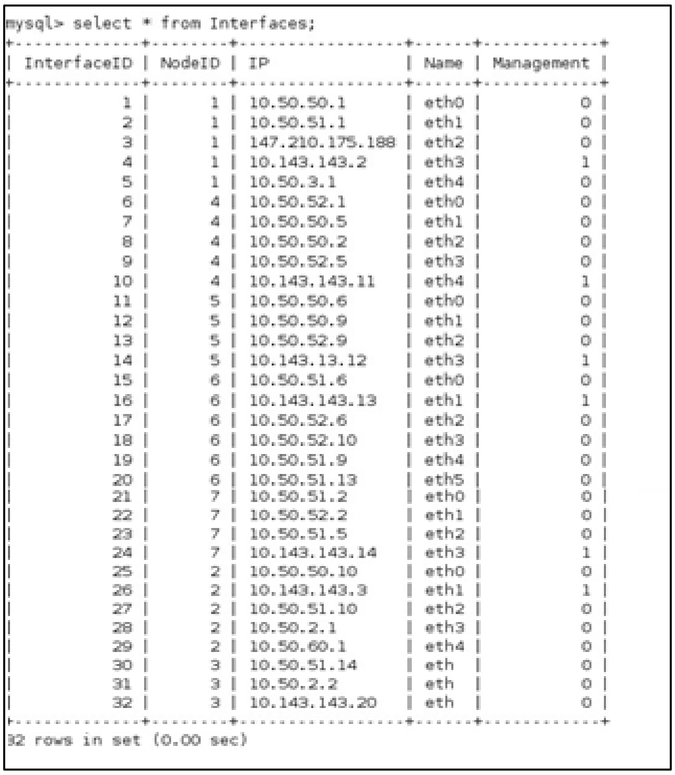
Step 1: LSP set up. Following the command of the Intra-NRM , information is retrieved from the network database (netDB) regarding the specifications of a respective LSP. Below is represented the information stored at the netDB. The LSP with ID 1000 has InterfaceIn id=1 and InterfaceOut id=25. This information is already installed in netDB and is retrieved whenever an a new LSP is required.



```
mysql> select * from LSPs;
+-------+------------+-------------+------------+------------+
| LSPID | InterfaceIn | InterfaceOut | SNMPAgentID | Bandwidth   |
+-------+------------+-------------+------------+------------+
|  1000 |           1 |           25 |           2 | 1.04858e+07 |
|  1001 |           1 |           25 |           1 | 1.04858e+07 |
|  1100 |           2 |           27 |           2 | 1.04858e+07 |
|  1200 |           1 |           27 |           2 | 1.04858e+07 |
|  2000 |          25 |            1 |        NULL | 1.04858e+07 |
|  2200 |          27 |            2 |        NULL | 1.04858e+07 |
|  2320 |          27 |           19 |           3 |        5000 |
|  5001 |          27 |            1 |        NULL | 1.04858e+07 |
+-------+------------+-------------+------------+------------+
8 rows in set (0.00 sec)
```

Figure 22. LSPs Table from netDB

The InterfaceIn id=1 corresponds to 10.50.50.1 eth0 (SW-MANE2) and the InterfaceOut id=25 to 10.50.50.10 eth0 (SW-MANE1)



```
mysql> select * from Interfaces;
+-------------+--------+-----------------+------+------------+
| InterfaceID | NodeID | IP              | Name | Management |
+-------------+--------+-----------------+------+------------+
|           1 |      1 | 10.50.50.1      | eth0 |          0 |
|           2 |      1 | 10.50.51.1      | eth1 |          0 |
|           3 |      1 | 147.210.175.188 | eth2 |          0 |
|           4 |      1 | 10.143.143.2    | eth3 |          1 |
|           5 |      1 | 10.50.3.1       | eth4 |          0 |
|           6 |      4 | 10.50.52.1      | eth0 |          0 |
|           7 |      4 | 10.50.50.5      | eth1 |          0 |
|           8 |      4 | 10.50.50.2      | eth2 |          0 |
|           9 |      4 | 10.50.52.5      | eth3 |          0 |
|          10 |      4 | 10.143.143.11   | eth4 |          1 |
|          11 |      5 | 10.50.50.6      | eth0 |          0 |
|          12 |      5 | 10.50.50.9      | eth1 |          0 |
|          13 |      5 | 10.50.52.9      | eth2 |          0 |
|          14 |      5 | 10.143.13.12    | eth3 |          1 |
|          15 |      6 | 10.50.51.6      | eth0 |          0 |
|          16 |      6 | 10.143.143.13   | eth1 |          1 |
|          17 |      6 | 10.50.52.6      | eth2 |          0 |
|          18 |      6 | 10.50.52.10     | eth3 |          0 |
|          19 |      6 | 10.50.51.9      | eth4 |          0 |
|          20 |      6 | 10.50.51.13     | eth5 |          0 |
|          21 |      7 | 10.50.51.2      | eth0 |          0 |
|          22 |      7 | 10.50.52.2      | eth1 |          0 |
|          23 |      7 | 10.50.51.5      | eth2 |          0 |
|          24 |      7 | 10.143.143.14   | eth3 |          1 |
|          25 |      2 | 10.50.50.10     | eth0 |          0 |
|          26 |      2 | 10.143.143.3    | eth1 |          1 |
|          27 |      2 | 10.50.51.10     | eth2 |          0 |
|          28 |      2 | 10.50.2.1       | eth3 |          0 |
|          29 |      2 | 10.50.60.1      | eth4 |          0 |
|          30 |      3 | 10.50.51.14     | eth  |          0 |
|          31 |      3 | 10.50.2.2       | eth  |          0 |
|          32 |      3 | 10.143.143.20   | eth  |          0 |
+-------------+--------+-----------------+------+------------+
32 rows in set (0.00 sec)
```

Figure 23. Interfaces Table from netDB

From the routes table we can also see the hops of the LSP 1000. That corresponds to SW-MANE2→LSR2→LSR1→SW-MANE1. A web representation of the network installation is also possible in order to have a visual representation of the installed configurations Figure 24.

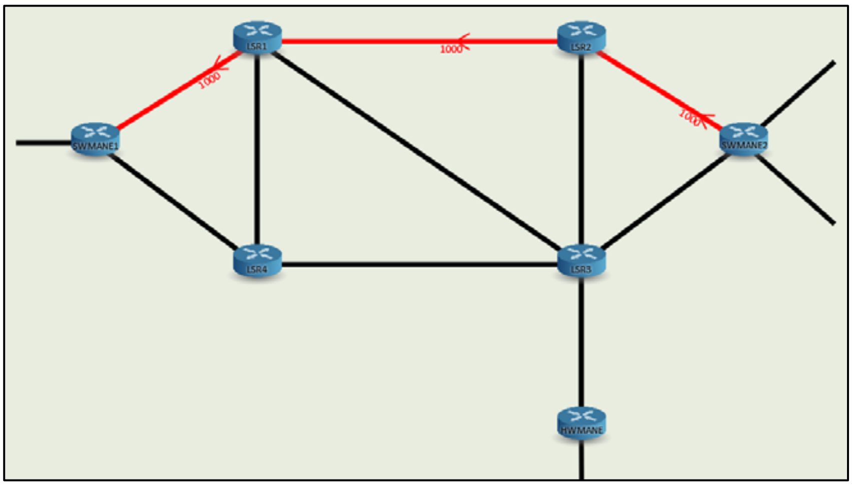
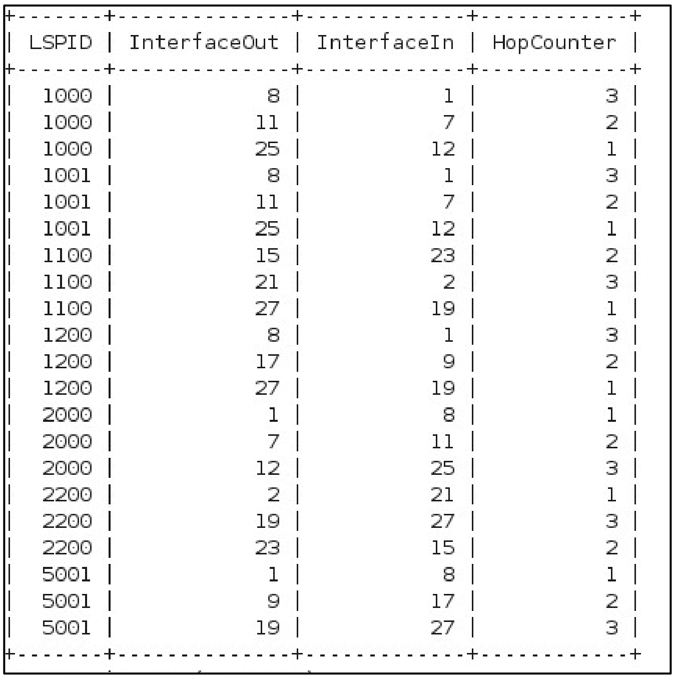


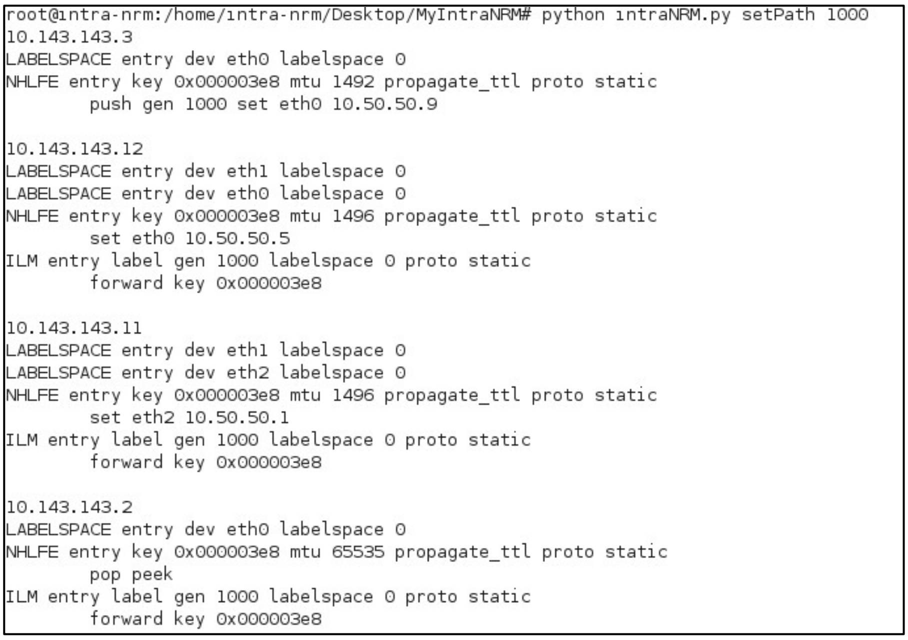
Figure 24. Web-interface of Intra-NRM/LSP representation



```
+-------+--------------+-------------+------------+
| LSPID | InterfaceOut | InterfaceIn | HopCounter |
+-------+--------------+-------------+------------+
|  1000 |            8 |           1 |          3 |
|  1000 |           11 |           7 |          2 |
|  1000 |           25 |          12 |          1 |
|  1001 |            8 |           1 |          3 |
|  1001 |           11 |           7 |          2 |
|  1001 |           25 |          12 |          1 |
|  1100 |           15 |          23 |          2 |
|  1100 |           21 |           2 |          3 |
|  1100 |           27 |          19 |          1 |
|  1200 |            8 |           1 |          3 |
|  1200 |           17 |           9 |          2 |
|  1200 |           27 |          19 |          1 |
|  2000 |            1 |           8 |          1 |
|  2000 |            7 |          11 |          2 |
|  2000 |           12 |          25 |          3 |
|  2200 |            2 |          21 |          1 |
|  2200 |           19 |          27 |          3 |
|  2200 |           23 |          15 |          2 |
|  5001 |            1 |           8 |          1 |
|  5001 |            9 |          17 |          2 |
|  5001 |           19 |          27 |          3 |
+-------+--------------+-------------+------------+
```

Figure 25. Routes Table from netDB

The path is configured and the Intra-NRM connects to the SW-MANEs at the edge of the network as well as at the core routers and creates the labelspace, the nhlfe and the ilm for each path. In Figure 26 we see the ingress/egress nodes of the configured LSP along with the transit nodes. We have created a NHLFE entry to add label 1000 and forward the packets to 10.50.50.9 using outgoing interface eth0. It is important to have set a labelspace so that the router expects MPLS packets through a specific interface.



```
root@intra-nrm:/home/intra-nrm/Desktop/MyIntraNRM# python intraNRM.py setPath 1000
10.143.143.3
LABELSPACE entry dev eth0 labelspace 0
NHLFE entry key 0x000003e8 mtu 1492 propagate_ttl proto static
        push gen 1000 set eth0 10.50.50.9

10.143.143.12
LABELSPACE entry dev eth1 labelspace 0
LABELSPACE entry dev eth0 labelspace 0
NHLFE entry key 0x000003e8 mtu 1496 propagate_ttl proto static
        set eth0 10.50.50.5
ILM entry label gen 1000 labelspace 0 proto static
        forward key 0x000003e8

10.143.143.11
LABELSPACE entry dev eth1 labelspace 0
LABELSPACE entry dev eth2 labelspace 0
NHLFE entry key 0x000003e8 mtu 1496 propagate_ttl proto static
        set eth2 10.50.50.1
ILM entry label gen 1000 labelspace 0 proto static
        forward key 0x000003e8

10.143.143.2
LABELSPACE entry dev eth0 labelspace 0
NHLFE entry key 0x000003e8 mtu 65535 propagate_ttl proto static
        pop peek
ILM entry label gen 1000 labelspace 0 proto static
        forward key 0x000003e8
```

Figure 26. LSP installation output with label 1000

Step 2: Second step is to configure the DiffServ traffic classes. We will support the following traffic classes: EF, AF11, AF12, AF21, AF22, BE. For AF we will use GRED qdiscs. In order to set up the GRED qdiscs, we used the following command:

```
#tc qdisc change dev eth0 parent 2:10 gred limit 4000k min 500k max 1000k burst 520
avpkt 1280 bandwidth 20000000 DP 1 probability 0.01 prio 1
```

The result with the configured classes will be as below:

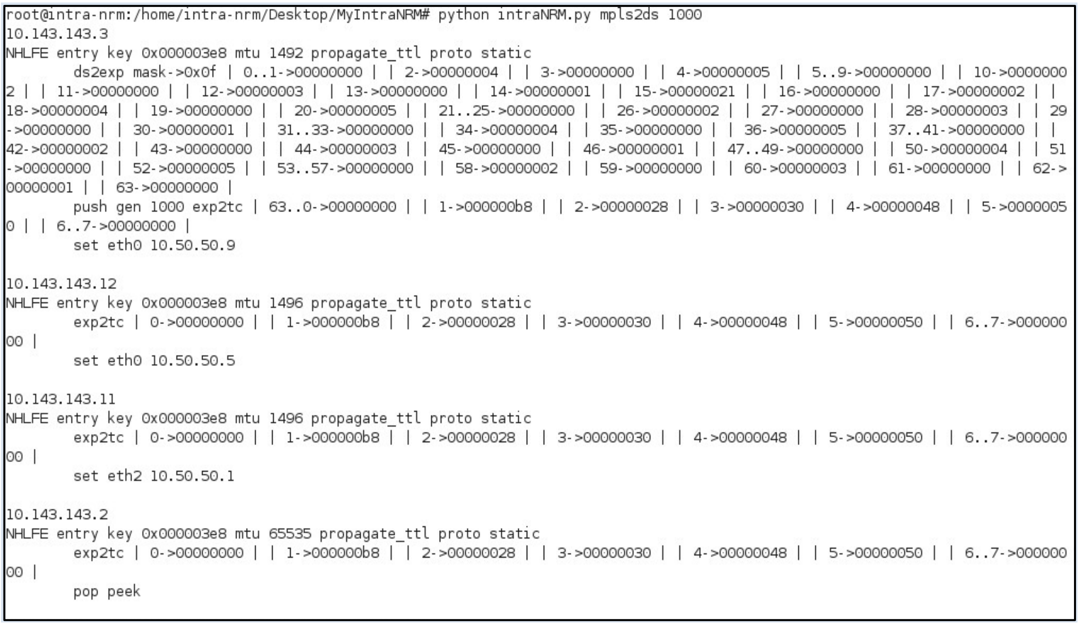
```
qdisc dsmark 1: root refcnt 6 indices 0x0040 default_index 0x0161 set_tc_index
 Sent 65017600 bytes 43499 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc htb 2: parent 1: r2q 10 default 0 direct_packets_stat 4
 Sent 65016098 bytes 43498 pkt (dropped 0, overlimits 465 requeues 0)
 backlog 0b 0p requeues 0
qdisc gred 5: parent 2:10
 DP:1 (prio 1) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 9893 (bytes 14859286)  ewma 8 Plog 26 Scell_log 15
 DP:2 (prio 2) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 9756 (bytes 14653512)  ewma 8 Plog 25 Scell_log 15
 Sent 29512798 bytes 19649 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc gred 6: parent 2:20
 DP:1 (prio 1) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 2947 (bytes 4426394)  ewma 8 Plog 25 Scell_log 15
 DP:2 (prio 2) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 1949 (bytes 2857234)  ewma 8 Plog 24 Scell_log 15
 Sent 7283628 bytes 4896 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
 qdisc pfifo 3: parent 2:50 limit 10p
  Sent 15219766 bytes 10133 pkt (dropped 0, overlimits 0
 requeues 0)
  backlog 0b 0p requeues 0
qdisc red 4: parent 2:60 limit 5033164b min 200Kb max 600Kb
 Sent 12993898 bytes 8816 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
  marked 0 early 0 pdrop 0 other 0
```

Step 3: Third step is to couple mpls with DiffServ functionality so that the mpls packets are forwarded in the correct queue based on their marking. For the MPLS network to function as a Diffserv network, two things must be done: the ingress nodes must make a mapping between the DSCP field in the IP packets and the exp field in the MPLS packet, and the core MPLS nodes must be able to offer differentiated services based on the previous markings. In our network, the following mapping has been taken into account:

Table 12. DSCP to EXP mapping

Class	DSCP	EXP
AF11	0x0a	2
AF12	0x0c	3
AF21	0x12	4
AF22	0x14	5
EF	0x2e	1
BE	0x00	0

In the following picture we see that the respective LSP coupled with DiffServ functionalities was correctly installed.

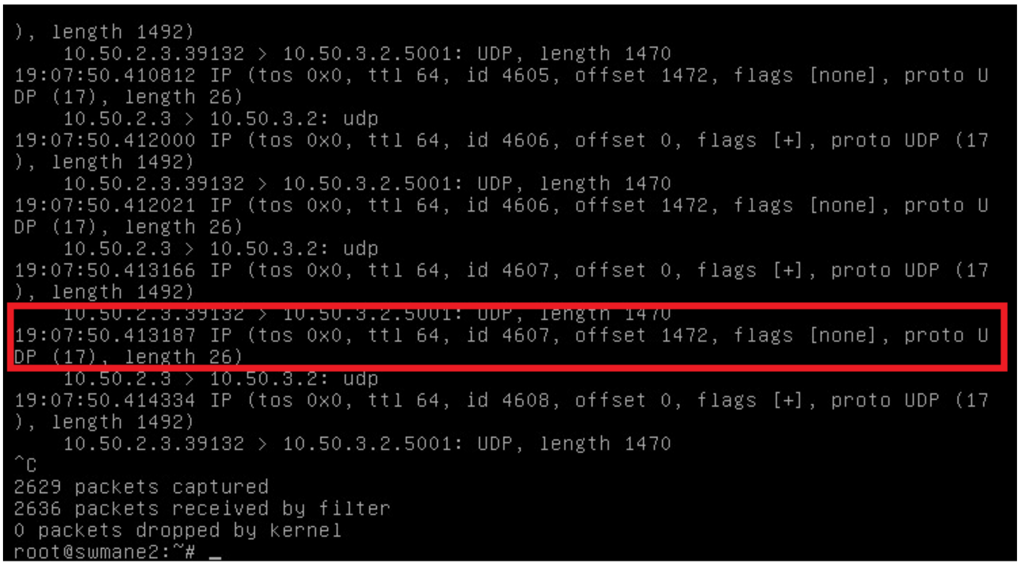


```
root@intra-nrm:/home/intra-nrm/Desktop/MyIntraNRM# python intraNRM.py mpls2ds 1000
10.143.143.3
NHLFE entry key 0x000003e8 mtu 1492 propagate_ttl proto static
        ds2exp mask->0x0f | 0..1->00000000 | | 2->00000004 | | 3->00000000 | | 4->00000005 | | 5..9->00000000 | | 10->0000000
2 | | 11->00000000 | | 12->00000003 | | 13->00000000 | | 14->00000001 | | 15->00000021 | | 16->00000000 | | 17->00000002 | |
18->00000004 | | 19->00000000 | | 20->00000005 | | 21..25->00000000 | | 26->00000002 | | 27->00000000 | | 28->00000003 | | 29
->00000000 | | 30->00000001 | | 31..33->00000000 | | 34->00000004 | | 35->00000000 | | 36->00000005 | | 37..41->00000000 | |
42->00000002 | | 43->00000000 | | 44->00000003 | | 45->00000000 | | 46->00000001 | | 47..49->00000000 | | 50->00000004 | | 51
->00000000 | | 52->00000005 | | 53..57->00000000 | | 58->00000002 | | 59->00000000 | | 60->00000003 | | 61->00000000 | | 62->
00000001 | | 63->00000000 |
        push gen 1000 exp2tc | 63..0->00000000 | | 1->000000b8 | | 2->00000028 | | 3->00000030 | | 4->00000048 | | 5->0000005
0 | | 6..7->00000000 |
        set eth0 10.50.50.9

10.143.143.12
NHLFE entry key 0x000003e8 mtu 1496 propagate_ttl proto static
        exp2tc | 0->00000000 | | 1->000000b8 | | 2->00000028 | | 3->00000030 | | 4->00000048 | | 5->00000050 | | 6..7->000000
00 |
        set eth0 10.50.50.5

10.143.143.11
NHLFE entry key 0x000003e8 mtu 1496 propagate_ttl proto static
        exp2tc | 0->00000000 | | 1->000000b8 | | 2->00000028 | | 3->00000030 | | 4->00000048 | | 5->00000050 | | 6..7->000000
00 |
        set eth2 10.50.50.1

10.143.143.2
NHLFE entry key 0x000003e8 mtu 65535 propagate_ttl proto static
        exp2tc | 0->00000000 | | 1->000000b8 | | 2->00000028 | | 3->00000030 | | 4->00000048 | | 5->00000050 | | 6..7->000000
00 |
        pop peek
```

Figure 27. Output after installing DiffServ with MPLS

In order to validate the settings above, we ran numerous experiments in Bordeaux Pilot. A simple scenario for the proof of concept is shown below. We sent UDP traffic from a user in AN3 (10.50.2.3) to a user in AN2 (10.50.3.2). The CAN manager requested the traffic to be marked as EF and to be forwarded in a specific VCAN pipe corresponding to LSP with label 1000.

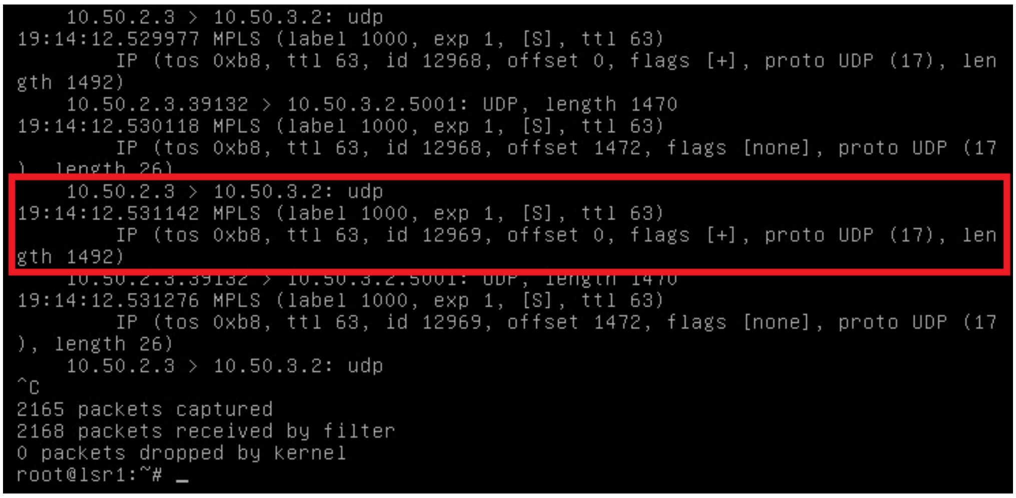
In order to validate the scenario we monitored the traffic step by step. In the ingress interface of MANE2 we checked that UDP traffic arrived from 10.50.2.3 →10.50.3.2 unmarked (TOS 0x0).



```
), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:07:50.410812 IP (tos 0x0, ttl 64, id 4605, offset 1472, flags [none], proto U
DP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:07:50.412000 IP (tos 0x0, ttl 64, id 4606, offset 0, flags [+], proto UDP (17
), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:07:50.412021 IP (tos 0x0, ttl 64, id 4606, offset 1472, flags [none], proto U
DP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:07:50.413166 IP (tos 0x0, ttl 64, id 4607, offset 0, flags [+], proto UDP (17
), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:07:50.413187 IP (tos 0x0, ttl 64, id 4607, offset 1472, flags [none], proto U
DP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:07:50.414334 IP (tos 0x0, ttl 64, id 4608, offset 0, flags [+], proto UDP (17
), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
^C
2629 packets captured
2636 packets received by filter
0 packets dropped by kernel
root@swmane2:~# _
```

Figure 28. Traffic trace MANE2 eth3

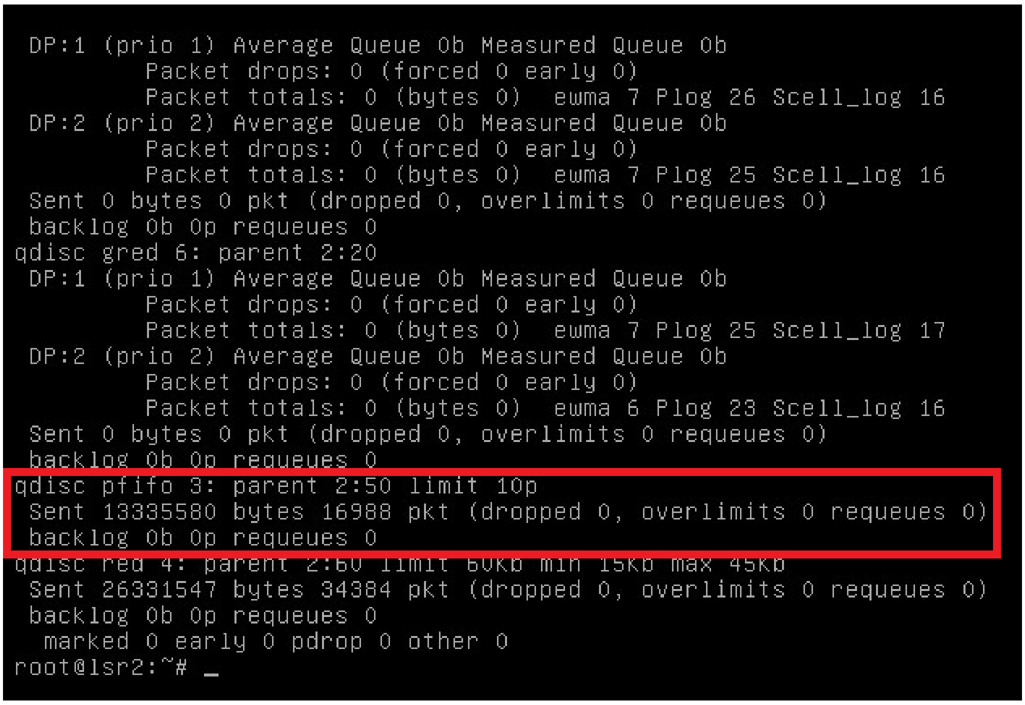
At the egress interface of MANE2 (eth0) the UDP traffic should be marked as EF and the packets should be encapsulated with label 1000.



```
    10.50.2.3 > 10.50.3.2: udp
19:14:12.529977 MPLS (label 1000, exp 1, [S], ttl 63)
        IP (tos 0xb8, ttl 63, id 12968, offset 0, flags [+], proto UDP (17), len
gth 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:14:12.530118 MPLS (label 1000, exp 1, [S], ttl 63)
        IP (tos 0xb8, ttl 63, id 12968, offset 1472, flags [none], proto UDP (17
), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:14:12.531142 MPLS (label 1000, exp 1, [S], ttl 63)
        IP (tos 0xb8, ttl 63, id 12969, offset 0, flags [+], proto UDP (17), len
gth 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:14:12.531276 MPLS (label 1000, exp 1, [S], ttl 63)
        IP (tos 0xb8, ttl 63, id 12969, offset 1472, flags [none], proto UDP (17
), length 26)
    10.50.2.3 > 10.50.3.2: udp
^C
2165 packets captured
2168 packets received by filter
0 packets dropped by kernel
root@lsr1:~# _
```

Figure 29. Traffic trace MANE2 eth0

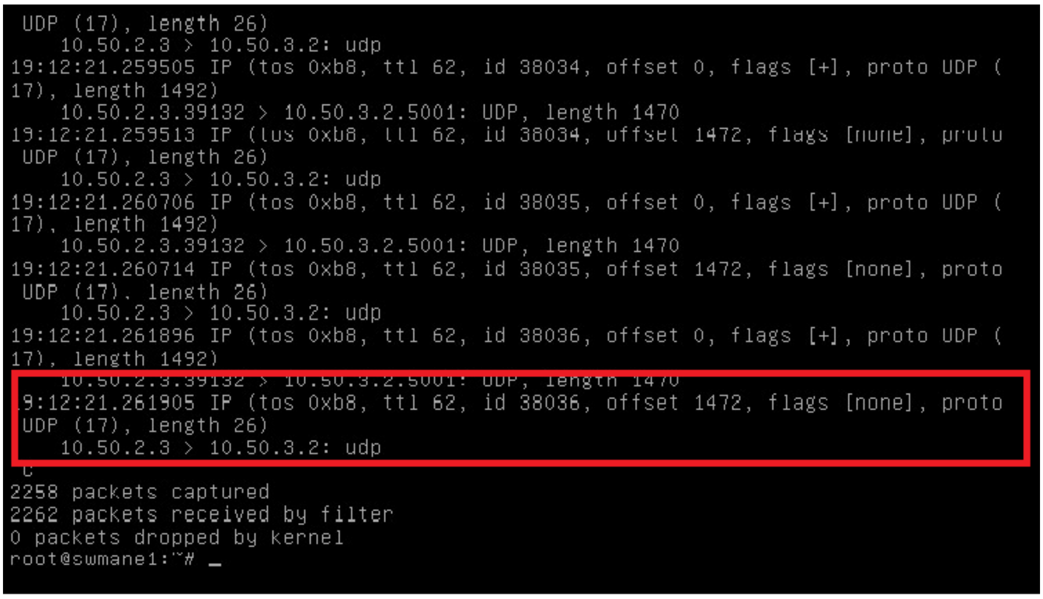
At the LSR2 (eth0), we verify that the EF traffic was forwarded from the correct queue.



```
 DP:1 (prio 1) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 0 (bytes 0)  ewma 7 Plog 26 Scell_log 16
 DP:2 (prio 2) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 0 (bytes 0)  ewma 7 Plog 25 Scell_log 16
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc gred 6: parent 2:20
 DP:1 (prio 1) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 0 (bytes 0)  ewma 7 Plog 25 Scell_log 17
 DP:2 (prio 2) Average Queue 0b Measured Queue 0b
        Packet drops: 0 (forced 0 early 0)
        Packet totals: 0 (bytes 0)  ewma 6 Plog 23 Scell_log 16
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc pfifo 3: parent 2:50 limit 10p
 Sent 13335580 bytes 16988 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc red 4: parent 2:60 limit 60Kb min 15Kb max 45Kb
 Sent 26331547 bytes 34384 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
  marked 0 early 0 pdrop 0 other 0
root@lsr2:~# _
```

Figure 30 Traffic trace in EF class queue LSR2 eth0

At last we checked that traffic arrived in MANE1 and at the egress interface the label was removed in order to reach the destination as ip packet.



```
 UDP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:12:21.259505 IP (tos 0xb8, ttl 62, id 38034, offset 0, flags [+], proto UDP (
17), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:12:21.259513 IP (tos 0xb8, ttl 62, id 38034, offset 1472, flags [none], proto
 UDP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:12:21.260706 IP (tos 0xb8, ttl 62, id 38035, offset 0, flags [+], proto UDP (
17), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:12:21.260714 IP (tos 0xb8, ttl 62, id 38035, offset 1472, flags [none], proto
 UDP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
19:12:21.261896 IP (tos 0xb8, ttl 62, id 38036, offset 0, flags [+], proto UDP (
17), length 1492)
    10.50.2.3.39132 > 10.50.3.2.5001: UDP, length 1470
19:12:21.261905 IP (tos 0xb8, ttl 62, id 38036, offset 1472, flags [none], proto
 UDP (17), length 26)
    10.50.2.3 > 10.50.3.2: udp
^C
2258 packets captured
2262 packets received by filter
0 packets dropped by kernel
root@swmane1:~# _
```

Figure 31. Traffic trace - label decapsulation SW-MANE1

5.3. Summary

This chapter described a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification, towards efficient resource exploitation. For this reason, it presented the elaboration of a prototype network management element: the Intra-NRM.

More specifically, it described the characteristics of the Intra-domain Network Resource Manager, in terms of a) monitoring the network layer resources, operation and status, b) controlling/configuring the resources inside the network domain c) delivering information regarding the underlying network status and dimensioning to the CAN Manager, prior to creating, terminating and/or modifying a VCAN.

Following the system's architecture, this chapter presented the implementation of a validation environment conformed to the design specifications, in order to verify the validity of the management network architecture based on content-aware and virtualization concepts, via a series of preliminary experiments. In this context, it presented the various mechanisms operated by the Intra-NRM towards optimally assigning network resources to virtual networks, based on requests from upper layers (e.g., CAN, Service layer). For this reason, a number of experiments were designed and conducted under controlled conditions. More specifically, we focused on the configuration of the network using MPLS and DiffServ technologies. Analysis of the experimental results verified the validity of the proposed architecture, establishing it as a novel solution in conjunction with the use of content-aware networks.

The work presented in this chapter has been reported in [158][159][160][161].

6. CONCLUSIONS

6.1. Overview

This final chapter of the thesis concludes it by resuming the research efforts, its scientific results and contribution to knowledge, as well as by identifying fields for future exploitation. In this context, section 6.2 summarises the work carried-out towards the proposal of a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification, assisted by a prototype Media-Aware Network Element (MANE) that offers content type recognition and content-based routing/forwarding as a matter of guaranteed QoS/QoE provision in an end-to-end approach. Section 6.3 elaborates on issues for future research based on the work carried out.

6.2. Thesis contribution to knowledge and research

The main contribution is centred on the proposal of a synergetic system architecture and the corresponding mechanisms that can optimally allocate the available network resources to diversified multimedia applications. Alongside, the appropriate network-layer interfaces have been proposed, allowing the underlying network to dynamically adapt its behavior according to changing network conditions. More specifically:

In Chapter 3 we presented a novel architecture (through the ALICANTE project -MediA Ecosystem Deployment through Ubiquitous Content-Aware Network Environments – European project) utilizing both Service and Network Awareness, besides moving one step beyond by converging them into a common operational system. The unifying factor in this consolidation process is Content/Service Awareness at the infrastructure level, i.e. the network capability to analyse media-flows and adjust its operation according to the content/service requirements for network available resources and the requested QoS/QoE policies.

In Chapter 4, a prototype content-aware network module (MANE), able to offer content type recognition and content-based routing/was presented. The main focus was put on the design of the MANE using a standard Linux kernel that encapsulates all the routing and forwarding capabilities available on a Linux platform and, at the same time, exploits content-aware functionalities using specific developed software modules. In order to manage and control the packet classification, queuing and scheduling processes needed to support traffic classification, QoS and MPLS, the developed software modules interface with the related user-space modules that natively control these processes. The result is that the content-aware modules are able to enforce certain policies and decisions based on the identification of the content type. Moreover, a specific signalling mechanism: the Content Aware Transport Information (CATI) header was proposed. This specific header is injected from service generation nodes in selected fields inside application layer protocol headers with the target to simplify the content identification process besides giving to the Service Provider the opportunity to request a more fine-grained classification of the handled flows. At last, the experimental results of the forwarding performance evaluation of the MANE were presented. The forwarding performance of the MANE is the same as the legacy IP router configuration almost 1Gbps (running on the same HW) for packets with frame sizes 256 bytes and higher. This is anticipated by the fact that the enhanced MANE functionalities, although they affect the CPU utilization, do not affect the forwarding performance. MANE during the timeframe needed for a classification and identification, does not intercept the packet-forwarding path through the Linux kernel structures, but copies the needed portions of data from the packet. When the identification has achieved its goal, the flow is automatically classified, accepting the fact that some (few) initial packets will be forwarded using the default policies.

In chapter 5, a synergetic management system capable to orchestrate cross-layer optimization processes for service differentiation/classification was described. To this end, it presented the architecture and the building blocks of a prototype network management element, the Intra-NRM, for monitoring the network layer resources, operation and status, controlling/configuring the resources inside the network domain and delivering information regarding the underlying network status to upper layers. Moreover, a validation environment conformed to the design specifications, in order to verify the validity of the management network architecture based on content-aware and virtualization concepts was described. In this context, various mechanisms operated by the Intra-NRM were tested through a number of experiments under controlled conditions. More specifically, we focused on the configuration of the network using MPLS and DiffServ technologies. Analysis of the experimental results verified the validity of the proposed architecture, establishing it as a novel solution when content-aware networks are used.

6.3. Fields for future exploitation

Several perspectives have been identified for future and further work. Regarding content-aware framework, although DPI methods appear to have good results in traffic classification, relying on the validation of the signature database, there are many cases where they cannot be applied. Especially in the case of encrypted traffic or in the case that payload parsing is not allowed to third parties, it becomes a challenge for any classification mechanism to classify the applications accurately. With encryption, all upper layer information becomes invisible to DPI mechanisms. Host behavior analysis methods can help among others to identify some applications. Statistical methods can also be used by exploiting traffic’s inherent characteristics, such as distribution of flow duration, flow idle time, packet inter-arrival time and packet lengths etc., which are considered to uniquely distinguish certain type of applications, with ultimate goal to cluster data traffic into groups that acquire similar traffic patterns [142].

Regarding intra-domain network resource management, the multi-path routing approach has been receiving a lot of attention as alternative methodology to ease traffic congestion and alleviate network resources’ consumption [143]. The basic idea of multi-path routing is to spread the packets across multiple alternate paths. The number of packets per path is decreased, so the emergence of bottlenecks is suppressed. Consequently, multipath routing aims to enhance various attributes such as the quality of service (QoS), delay, and delivery reliability [144]. An interesting approach is based on Multinomial Logit Model (MNL) that focuses on the calculation of the utility that a corresponding packet will inherit by choosing a specific path, aiming not only at the reduction of packet accumulation on some links but also at the bandwidth availability increase in the network [145]. Unlike existing multipath routing schemes [146], which pre-set alternate paths [147][148], this method can dynamically distribute packets to every possible path and thus may be more efficient especially if combined with content awareness concepts (i.e., the ability to know a priori the content of a packet)[149].

At last, further mechanisms resided in the Intra-NRM that would allow dynamic content-aware routing can be presented relying on the hypothesis that no other previous configurations exist. Intra-NRM will be responsible for the efficient network resource allocation process that will ensure minimum network bandwidth’s segmentation across the predefined VCANS. To achieve these, the Intra-NRM may be enabled to exploit optimization methods [150], among which are the decision-making ones that will try to reach an optimal solution through classical mathematical rationalization, i.e. by formulating an objective function so that equality and inequality constraints are not crossed. Such decision-making mechanisms may be implemented, through a number of optimisation techniques, such as the integer/combinatorial programming (e.g. Backtracking) and the mathematical programming (e.g. Simulated Annealing). It should be noted, however, that the choice of the most appropriate decision-making mechanism implementation technique constitutes an application-driven approach, based on specific use-case scenarios, and by taking into account the corresponding implementation intricacies. Thereupon, metrics such as the complexity of the algorithm, the range of the possible solutions to be checked, the processing time and computational power required for obtaining the optimum solution have to be considered prior to choosing the most applicable technique.

APPENDIX A

List of acronyms

	Acronym	Description
A	AN	Access Network
	ANP	Access Network Provider
	API	Application Programming Interface
	AS	Autonomous system
	AIT	Advanced IPTV Terminal
	AVC	Advanced Video Coding
B	BL	Base Layer
	BW	Bandwidth
C	CAN	Content-Aware Network
	CATI	Content-Aware Transport Information
	CCN	Content-Centric Networking
	CON	Content-Oriented Networking
	CANP	Content-Aware Network Provider
	CC	Content Creator
	CP	Content Provider
	CPl	Control Plane
	CPE	Customer Premises Equipment
	CPU	Central Processing Unit
	CS	Content server
D	DB	Database
	DPl	Data Plane
E	E2E	End To End
	EU	End-User
	EUT	End-User Terminal
F	FI	Future Internet
	FMIA	Future Media Internet Architecture

G	GUI	Graphical User Interface
H	HB	Home-Box
	HBDB	Home-Box Database
	HD	High Definition
	HG	Home Gateway
	HGI	Home Gateway Initiative
	HTTP	HyperText Transfer Protocol
I	IETF	Internet Engineering Task Force
	IMS	IP Multimedia Subsystem
	IP	Internet Protocol
	IPTV	Internet Protocol Television
M	MANE	Media-Aware Network Element
	MPEG	Moving Picture Experts Group
	MPl	Management Plane
	MPLS	Multi-Protocol Label Switching
N	NAA	Network-Aware Application
	NE	Network Environment
	NIA	Network Interconnection Agreement
	NP	Network Provider
	NRM	Network Resource Management
O	OTT	Over-The-Top
	OSI	Open Systems Interconnection
P	P2P	Peer to Peer
Q	QoE	Quality Of Experience
	QoS	Quality Of Service
R	RTP	Real Time Transport Protocol
S	SE	Service Environment
	SDN	Software Defined Networking
	SIP	Session Initiation Protocol
	SLA	Service Level Agreement
	SON	Service Oriented Architecture
	SOA	Service Oriented Architecture
	SP	Service Provider
	SR	Service Registry
	STB	Set-Top Box
U	UDP	User Datagram Protocol

	UGC	User-Generated Content
	UE	User Environment
	UP	User Profile
	URL	Uniform Resource Locator
	VC	Video Conference
V	VCAN	Virtual CAN
	VoD	Video On Demand
	VM	Virtual Machine
W	WiFi	Wireless Fidelity
X	XML	Extensible Markup Language

APPENDIX B

Publications

B1. Published papers in international conferences and workshops

1. G. Gardikis, G. Xilouris, D. Negru, **P. Anapliotis**, Y. Chen, E. Pallis, A. Kourtis, “Media Ecosystem Deployment in a Content-Aware Future Internet Architecture”, in proc. of the IEEE Symposium on Computers and Communications 2011 (ISCC 2011), Corfu, Greece, June 28th-July 1st, 2011.
2. **P. Anapliotis**, D. Negru, E. Pallis, V. Zacharopoulos, “Enhancing Legacy Infrastructures with Content Aware Enablers towards A Networked-Media Platform”, in proc. of the IEEE International Conference on Multimedia and Expo (ICME 2011), Barcelona, Spain, July 11-15, 2011.
3. **P. Anapliotis**, E. Markakis, A. Sideris, E. Pallis, D. Negru, “A novel content-aware multipath routing concept exploiting random utility theory principles”, in proc. of the IEEE international conference on Telecommunications and Multimedia (TEMU 2012), Heraklion, Crete, Greece, 30 July-1 August, 2012.
4. E. Markakis, A. Sideris, **P. Anapliotis**, G. Alexiou, C. Skianis, E. Pallis, “IMS-enabled interactive broadcasting network utilizing peer to peer constellations”, in proc. of the IEEE international conference on Telecommunications and Multimedia (TEMU 2012), Heraklion, Crete, Greece, 30 July-1 August, 2012.
5. A. Sideris, E. Markakis, **P. Anapliotis**, E. Pallis, C. Skianis, “Content Adaptation of IPTV Services in Interactive DVB-T systems”, in proc. of the IEEE international conference on Telecommunications and Multimedia (TEMU 2012), Heraklion, Crete, Greece, 30 July-1 August, 2012.

B2. Main research projects Deliverables

1. G. Xilouris, G. Gardikis (eds.) et al., ICT ALICANTE, Deliverable D2.1: "ALICANTE Overall System and Components Definition and Specification", September 2011, http://www.ict-alicante.eu/validation/download/work-package/alicante_d2.1_final.pdf.
2. M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D2.3: "Cross-layer Monitoring Definition and Specification", September 2011, http://www.ict-alicante.eu/validation/download/work-package/alicante_d2.3_final.pdf.

3. R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D2.4: "Definition and Specification of the ALICANTE Pilot Architecture", September 2011, http://www.ict-alicante.eu/validation/download/work-package/ alicante_ d2.4_final.pdf .
4. M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D4.1.1: "The ALICANTE Home-Box Layer – I", September 2011, http://www.ict-alicante.eu/validation/download/work-package/alicante_d4.1.1_final.pdf .
5. R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D4.2.1: "Home-Box Layer Interfaces and Monitoring – I", September 2012, http://www.ict-alicante.eu/validation/download/work-package/alicante_d4.2.1_final.pdf .
6. E. Markakis, (ed) et al., ICT ALICANTE, Deliverable D4.3.1: " Home-Box Layer Services – I", September 2012, http://www.ict-alicante.eu/validation/download/work-package/alicante_d4.3.1_final.pdf .
7. M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D4.4: " The ALICANTE Home-Box Layer – Final", May 2013, http://www.ict-alicante.eu/validation/download/work-package/alicante_d4f_v1.0.pdf .
8. A.Mevel (ed) et.al, ICT ALICANTE, Deliverable D6.1.1: "Content-Aware Network Infrastructure and Elements – Intermediate", September 2011, http://www.ict-alicante.eu/validation/download/work-package/alicante_ d6.1.1_ final.pdf .
9. E.Borcoci (ed.) et. al. ICT ALICANTE D6.2.1: "CAN Management, Control and Interfaces –I", February 2012, http://www.ict-alicante.eu/validation/download/work-package/alicante_d6.2.1_final.pdf .
10. E.Pallis (ed.) et al., ICT ALICANTE D6.3.1: "Network Layer Management and Control", July 2012, http://www.ict-alicante.eu/validation/download/work-package/alicante_d6.3.1_final.pdf.
11. E.Borcoci (ed.) et. al. ICT ALICANTE D6.6: "The ALICANTE CAN and Network Environment", June 2013, http://www.ict-alicante.eu/validation/download/work-package/alicante_d6f_v1.0.pdf .
12. R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D8.2: " The ALICANTE Pilot Integration – Final", September 2013, http://www.ict-alicante.eu/validation/download/work-package/alicante_d8.2_v1.1.pdf .
13. Y. Zhang (ed) et al., ICT ALICANTE, Deliverable D8.3: "Trials and Validation", September 2013, http://www.ict-alicante.eu/validation/download/work-package/alicante_d8.3_v1.1.pdf.

REFERENCES

[1] V. Jacobson et al., “Networking Named Content”, Proc. ACM CoNEXT 2009. Rome, Italy, December 2009

[2] B. Subbiah, Z. Uzmi, “Content aware networking in the Internet: issues and challenges”, Proc. IEEE Int. Conf on Communications, 2001, vol.4, no., pp.1310-1315, doi:10.1109/ICC.2001.936912

[3] Cheok, Adrian David, “Art and Technology of Entertainment Computing and Communication”, ISBN 978-1-84996-136-3. Springer-Verlag London Limited, 2010, pp. 59

[4] W. K.Chai, Ning Wang, I.Psaras G.Pavlou, C.Wang, G. García de Blas, F.J. Ramon Salguero, Lei Liang, S.Spirou, Andrzej Beben, E. Hadjioannou, “CURLING: Content- Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services” , IEEE Communications Magazine, March 2011, pp.112-120

[5] 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, pp. 121-127

[6] V. Jacobson et al., “Networking Named Content”, Proc. ACM CoNEXT 2009. Rome, Italy, December 2009

[7] R. D. Callaway, A. Rodriguez, M. Devetsikiotis, and G. Cuomo. "Challenges in Service-Oriented Networking", Proceedings of the 49th Annual IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, November 2006.

[8] A. Galis et. al., “Management and Service-aware Networking Architectures (MANA) for Future Internet Position Paper: System Functions, Capabilities and Requirements”, http://www.future-internet.eu/home/future-internet-assembly/prague-may-2009.

[9] Theodore Zahariadis, Dimitri Papadimitriou, Hannes Tschofenig, Stephan Haller, Petros Daras, George D. Stamoulis, and Manfred Hauswirth. “Towards a future internet architecture” The future internet, John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, and Dave Lambert (Eds.). Springer-Verlag, Berlin, Heidelberg 7-18.

[10] S. Paul, J. Pan, & R. Jain, “Architectures for the future networks and the next generation Internet: A survey”, Computer Communications, Volume 34, Issue 1, 15 January 2011, Pages 2-42.

[11] W. Chai, et al, “Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services”, IEEE Communications Magazine, vol. 49, no. 3, pp. 112-120.

[12] R. Lloyd Cody, E. Tsekleves and J. Cosmas “Open-standards Rich Media Mobile Platform & Rapid Deployment Service Creation Tool” World Wireless Congress 2008, May 14 - 16, 2008, San Francisco Bay Area, California.

[13] Seamless User-Generated Content Sharing in the Extended Home (2009). Faculty of Computing and Electrical Engineering, Tampere University of Technology Finland.

[14] The Downside of User-generated Content", BusinessWeek.com, Mar 2010.

[15] Service/content creation, publication and dissemination limitations", White Paper, Infosys, http://www.infosys.com/offerings/industries/communication-services/white-papers/Documents/realizing-potential-user-generated.pdf.

[16] M. Cha, H. Kwak, P. Rodriguez, Y.Y Ahn and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system" Proc. ACM SIGCOMM Conference on Internet Measurement (IMC), 2007, 1-14.

[17] R. Lobato et al, "Histories of User-Generated Content: Between Formal and Informal Media Economies", Int. J. of Communications, Vol.5 (2011).

[18] K. J. Ma, R. Bartoš, and S. Bhatia, "A Survey of Schemes for Internet-based Video Delivery," Elsevier Journal of Network and Computer Applications, 15 p., February 2011.

[19] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/AVC Standard", Proceedings of the IEEE, vol. 99, 2011, pp. 626-642.

[20] A. Vetro, A. Tourapis, K. Muller, and T. Chen, "3D-TV content storage and transmission", IEEE Transactions on Broadcasting, vol. 57, 2011, pp. 384-394.

[21] "Achieving quality IPTV over DSL", White Paper , Broadband forum, Marketing Group, 2010.

[22] "Next Generation Broadband Access", White Paper, Broadband forum, Marketing Group, 2010.

[23] N. Ling, "Expectations and Challenges for Next Generation Video Compression", Proc. 5th IEEE Conf. on Industrial Electronics and Applications, 2010.

[24] Marie-José Montpetit, Natalie Klym, and Emmanuel Blain, "The Future of Mobile TV: When Mobile TV Meets the Internet and Social Networking", Book "Mobile TV: Customizing Content and Experience, Human-Computer Interaction Series", ISBN 978-1-84882-702-8. Springer-Verlag London, 2010, pp. 305.

[25] J. Leblet, Z. Li, G. Simon, and D. Yuan,"Optimal network locality in distributed virtualized data-centres", Computer communications, 2011 (Accepted for publication).

[26] K. El-Khatib, G. Bochmann, A. Saddik, "On the Use of Web Services in Content Adaptation", ISBN 978-1-60566-330-2. University of Ontario Institute of Technology, 2009. Chapter VIII.

[27] Surendar Chandra, Ashish Gehani, Carla Schlatter Ellis, Amin Vahdat, "Transcoding characteristics of Web images", In ACM/SPIE Multimedia Computing and Networking, 2001.

[28] Dey, A.K. Abowd, G.D. "Towards a Better Understanding of Context and Context-Awareness", CHI 2000, Workshop on the What, Who, Where, When, and How of Context-Awareness, 2000.

[29] Th. Zahariadis, F. Junqueira, L. Celetto, E. Quacchio, S. Niccolini, P. Plaza, "Content aware searching, caching and streaming," 2nd International Conference on Telecommunications & Multimedia, Chania, Greece, 14-16 July 2010, pp. 263-265.

[30] Mostafa Zaman Chowdhury, Bui Minh Trung, Yeong Min Jang, Young-Il Kim and Won Ryu, "Service Level Agreement for the QoS Guaranteed Mobile IPTV Services over Mobile WiMAX Networks", The Journal of Korea Information and

Communications Society (KICS), vol. 36, no. 4, April 2011, Chapter “III. Proposed SLA Architecture”.

[31] ETSI ETR 266 Methods for Testing and Specification (MTS); Test Purpose style guide, August 1996.

[32] Microsoft Mediaroom IPTV Service, http://www.microsoft.com/mediaroom/, Employed in many work-class operations such as, at&t, Bell, Deutsche Telekom, Swisscom, Portugal Telecom, Hol, TDC, GZCATV.

[33] ETSI TS 182 027 V8.8.0 “Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), IPTV Architecture, IPTV functions supported by the IMS subsystem”, February 2008.

[34] ETSI DTS 02049 V0.0.8, tech. spec. draft, “IPTV Architecture: Dedicated Subsystem for IPTV Functions in NGN” Sept. 2007.

[35] Open IPTV Forum, “Service and Platform Requirements”, “Services and Functions”, “Functional Architecture, Solution Specification” – volumes 1-7, http://www.openiptvforum.org/.

[36] Dale Schmidt and Don Kamarga, “Economic Drivers for IMS-based Converged Services”, Siemens Networks LLC white paper, 2006, Chapter “The IMS Value Proposition”, http://www.siemensnetworksmedia.com/Whitepapers/Economic_Drivers_for_IMS.pdf.

[37] Eugen Mikoczy, Dmitry Sivchenko, Bangnan Xu and Jose I. Moreno, “IPTV Systems, Standards and Architectures: Part II-IPTV Services over IMS: Architecture and Standardization”, IEEE Communications Magazine, Volume 46, 2008, Chapter “Introduction”.

[38] Wolfgang Theilmann and Luciano Baresi, “Multi-Level SLAs for Harmonized Management in the Future Internet”, Book “Towards the Future Internet - A European Research Perspective”, ISBN 978-1-60750-007-0, 2009.

[39] George Pallis and Athena Vakali, “Insight and Perspectives for Content Delivery Networks”, Communications of the ACM (CACM), vol. 49, 2006, pp. 101-106.

[40] Adrian Popescu, Demetres D. Kouvatsos, David Remondo and Stefano Giordano, “Content distribution over IP: Developments and challenges”, Book “Network performance engineering”, ISBN: 978-3-642-02741-3, 2011, Chapter “3- Content Distribution Networks”.

[41] M. Freedman, “Experiences with CoralCDN: A Five-Year Operational View,” Proc. 7th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '10) San Jose, CA, May 2010.

[42] “AKAMAI to enable Web for DVD and HD video”, August 31, 2007, http://www.akamai.com/dl/akamai/Akam_in_Online_Reporter.pdf.

[43] G. Sakaryan and H. Unger, “Topology Evolution in P2P Distributed Networks”, IASTED: Applied Informatics, 2003.

[44] Matthew Roughan, “Robust Network Planning”, of the Guide to Reliable Internet Services and Applications, Springer, 2010, Chapter I, http://www.maths.adelaide.edu.au/matthew.roughan/Papers/Ch5-RobustPlanning.pdf.

[45] “Hyperconnectivity and the Approaching Zettabyte Era”, White Paper, Cisco Visual Networking Index(VNI), Jun 02, 2010,

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.html.

[46] M. Buchanan, “10 Percent of Broadband Subscribers Suck up 80 Percent of Bandwidth But P2P No Longer To Blame,” Gizmodo, 22 April, http://gizmodo.com/382691/10-percent-of-broadband-subscriber-suck-up-80-percent-of-bandwidth-but-p2p-no-longer-to-blame.

[47] BitTorrent, http://www.bittorrent.com.

[48] Oussama Layaida, Slim Benatallah, Daniel Hagimont, “A Framework for Dynamically Configurable and Reconfigurable Network-based Multimedia Adaptations”, “Journal of Internet Technology”, 2004, pp. 57-59.

[49] Xin Wang and Henning Schulzerinne, “Comparison of Adaptive Internet Multimedia Applications”, Institute of Electronics, Information and Communication Engineers Transactions , vol. E82-B, , June 1999, pp. 806–818.

[50] W. Chai, et al, “Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services”, IEEE Communications Magazine, vol. 49, no. 3, pp. 112-120.

[51] R. Lloyd Cody, E. Tsekleves and J. Cosmas “Open-standards Rich Media Mobile Platform & Rapid Deployment Service Creation Tool” World Wireless Congress 2008, May 14 - 16, 2008, San Francisco Bay Area, California.

[52] “The Downside of User-generated Content”, BusinessWeek.com, Mar 2010.

[53] FIArch Group: Fundamental Limitations of Current Internet and the path to Future Internet (March 2011), http://ec.europa.eu/information_society/activities/foi/docs/current_internet_limitations_v9.pdf.

[54] Internet Architecture Task Force, “Internet Architecture for Innovation”, December 2010, http://ec.europa.eu/information_society/activities/foi/library/docs/p3-aiai-2010.pdf.

[55] Future Media Internet Architecture Think Tank (FMIA-TT), “Future Media Internet Architecture Reference Model”, 1. March 2011, http://www.future-internet.eu/uploads/media/FMIA_Reference_Architecture.pdf.

[56] Ripeanu, M. (2001, August). Peer-to-peer architecture case study: Gnutella network. In Peer-to-Peer Computing, 2001. Proceedings. First International Conference on (pp. 99-100). IEEE.

[57] Clarke, I., Sandberg, O., Wiley, B., & Hong, T. W. (2001, January). Freenet: A distributed anonymous information storage and retrieval system. In Designing Privacy Enhancing Technologies (pp. 46-66). Springer Berlin Heidelberg.

[58] N. M. K. Chowdhury and R. Boutaba, “Network virtualization: State of the art and research challenges,” IEEE Communications Magazine, vol. 47, no. 7, pp. 20–26, July 2009

[59] 4WARD Project, http://www.4ward-project.eu/

[60] Y. Zu, R. Zhang-Shen, S.Rangarajan, and J. Rexford, “Cabernet: Connectivity Architecture for Better Network Services”, in Proc. ACM ReArch ’08, Madrid, Spain, December 2008.

[61] GEYSERS: Generalised Architecture for Dynamic Infrastructure Services, http://www.geysers.eu/

[62] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A.Wundsam, M. Kind, O. Maennel, and L. Mathy, “Network Virtualization Architecture: Proposal and Initial Prototype”, in Proc. ACM SIGCOMM VISA, Barcelona, Spain, August 2009.

[63] C. Werle, P. Papadimitriou, I. Houidi, W. Louati, D. Zeghlache, R. Bless, and L. Mathy, “Building Virtual Networks Across Multiple Domains”, in Proc. ACM SIGCOMM 2011, Poster Session, Toronto, Canada, August 2011

[64] P. Papadimitriou, I. Houidi, W. Louati, D. Zeghlache, C. Werle, R. Bless, and L. Mathy, “Towards Large-Scale Network Virtualization”, IFIP WWIC 2012, Santorini, Greece, June 2012 (Best Paper Award)

[65] E. Abarca, J. Grassler, G. Schaffrath, S. Schmid, “A Federated CloudNet Architecture: The PIP and the VNP Role”, arXiv:1303.6753v1 *cs.NI+, March 2013

[66] J. Nogueira, M. Melo, J. Carapinha, S. Sargento, “A platform for Operator-driven Network Virtualization”, Proc. IEEE EUROCON - International Conference on Computer as a Tool 2011, pp.1-4

[67] S. Zhang, Z. Qian, S. Guo, S. Lu, “FELL: A Flexible Virtual Network Embedding Algorithm with Guaranteed Load Balancing”, in Proc. 2011 IEEE International Conference on Communications (ICC), 5-9 June 2011

[68] S.B. Masti, S.V. Raghavan, “VNA: An Enhanced Algorithm for Virtual Network Embedding”, in 21st International Conference on Computer Communication Networks (ICCCN), July 30 - August 02 2012

[69] M. Chowdhury, M. R. Rahman, R. Boutaba, “ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping”, 206 IEEE/ACM Transactions on Networking, Vol. 20, No. 1, February 2012

[70] I. Houidi, W. Louati, W. Bean-Ameur, and D. Zeghlache, “Virtual Network Provisioning Across Multiple Substrate Networks”, Computer Networks, Vol. 55, No. 4, March 2011.

[71] J. Nogueira, M. Melo, J. Carapinha, S. Sargento, “Network Virtualization System Suite: Experimental Network Virtualization Platform”, Proc. International Conf. on Testbeds and Research Infrastructures for the development of Networks and Communities, Shanghai, China, April , 2011

[72] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R.Neugebauer, I.Pratt, and A.Warfield, “Xen and the Art of Virtualization”, in Proc. 19th ACM Symposium on OS Principles, Bolton Landing, NY, USA, Oct. 2003.

[73] E. Rosen and Y. Rekhter: BGP/MPLS VPNs, RFC 2547, IETF, March 1999.

[74] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Generic Routing Encapsulation (GRE)”, RFC2784, IETF, March 2000.

[75] E. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, “Towards High Performance Virtual Routers on Commodity Hardware”, in Proc. ACM CoNEXT 2008, Madrid, Spain, December 2008.

[76] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, P. Papadimitriou, “A platform for high performance and flexible virtual routers on commodity hardware”, ACM SIGCOMM Computer Communication Review archive, vol.40, no.1, pp.127-128, January 2010

[77] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI Veritas: Realistic and Controlled Network Experimentation", in Proc. ACM SIGCOMM, Pisa, Italy, September 2006.

[78] S. Bhatia, M. Motiwala, W. Muhlbauer, Y. Mundada, V. Valancius, A. Bavier, N. Feamster, L. Peterson, and J. Rexford, "Trellis: A Platform for Building Flexible, Fast Virtual Networks on Commodity Hardware", in Proc. 3rd ACM Workshop on Real Overlays and Distributed Systems, Madrid, Spain, December 2008.

[79] J. Carapinha, J. Jimenez, "Network Virtualization – a View from the Bottom", in Proc. ACM SIGCOMM VISA'2009 Workshop, Barcelona, 17 August 2009

[80] N. McKeown. Keynote talk: Software-defined networking. In Proc. of IEEE INFOCOM'09,Apr. 2009.

[81] SDN Technical White Paper http://h17007.www1.hp.com/docs/interopny/4AA4-3871ENW.pdf.

[82] HP SDN/Openflow Technology Solutions http://h17007.www1.hp.com/us/en/solutions/technology/openflow/index.aspx?jumpid=in_r11652_us/en/openflow-114x110/solutions/banner.

[83] SDN Controller Product Fact Sheet: http:h17007.www1.hp.com/docs/interopny/4AA4-3881ENW.PDF.

[84] "OpenFlow: Enabling Innovation in Campus Networks"- Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner.

[85] B.Raghavan, T.Koponen, A.Ghodsi, M.Casado, S.Ratnasamy, S.Shenker, Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure, SIGCOMM 2012, http://conferences.sigcomm.org/hotnets/2012/papers/hotnets12-final76.pdf

[86] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox. Intelligent Design Enables Architectural, Evolution. In Proc. of Hotnets-X, 2011.

[87] T. Koponen, S. Shenker, S. Shenker, H. Balakrishnan, N. Feamster, I. Ganichev, A. Ghodsi, P. B. Godfrey, N. McKeown, G. Parulkar, B. Raghavan, J. Rexford, S. Arianfar, and D. Kuptsov. Architecting for Innovation. SIGCOMM CCR, 41(3), 2011.

[88] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian, Fabric: A Retrospective on Evolving SDN. In Proc. Of HotSDN, August 2012.

[89] G. Gardikis, G. Xilouris, D. Negru, P. Anapliotis, Y. Chen, E. Pallis, A. Kourtis "Media Ecosystem Deployment in a Content-Aware Future Internet Architecture" accepted to be published in the Sixteenth IEEE Symposium on Computers and Communications 2011 (IEEE ISCC 2011), Corfu, Greece, June 28th-July 1st 2011

[90] Theodore Zahariadis, Dimitri Papadimitriou, Hannes Tschofenig, Stephan Haller, Petros Daras, George D. Stamoulis, and Manfred Hauswirth. "Towards a future internet architecture" The future internet, John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, and Dave Lambert (Eds.). Springer-Verlag, Berlin, Heidelberg 7-18

[91] 4WARD, "A clean-slate approach for Future Internet", http://www.4ward-project.eu/

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

[93] Gerardo García de Blas, (Ed) et al., Public Deliverable D2.1, “Business Models and System Requirements for the COMET System”, 2011, http://www.comet-project.org/

[94] Publish-Subscribe Internet Routing Paradigm, http://www.psirp.org/ M. Ain ed. et. al., Public Deliverable D2.2, “Conceptual Architecture of PSIRP“ http://www.psirp.org/

[95] Wei Koong Chai, Ning Wang, I. Psaras, G. Pavlou, Chaojiong Wang, G. G. de Blas, F. J. Ramon-Salguero, Lei Liang, S. Spirou, A. Beben, E. Hadjioannou, "Curling: Content ubiquitous resolution and delivery infrastructure for next-generation services, "Communications Magazine, IEEE, vol.49, no.3, pp.112-120, March 2011

[96] N. Ramzan, E. Quacchio, T. Zgaljic, S. Asioli, L. Celetto, E. Izquierdo, F. Rovati, "Peer-to-peer streaming of scalable video in future Internet applications," Communications Magazine, IEEE, vol.49, no.3, pp.128-135, March 2011

[97] H. Koumaras, D. Négru et al., “Media Ecosystems: A Novel Approach for Content Awareness in Future Networks”, Future Internet: Achievements and Promising Technology, Springer Verlag, pp.369-380. May 2011

[98] FIArch Group: Fundamental Limitations of Current Internet and the path to Future Internet (March 2011), http://ec.europa.eu/information_society/activities/foi/docs/current_internet_limitations_v9.pdf

[99] Future Media Internet Architecture Think Tank (FMIA-TT), “Future Media Internet Architecture Reference Model”, 1. March 2011, http://www.futureinternet.eu/uploads/media/FMIA_Reference_Architecture.pdf

[100] Howarth, M.P. et al., “Provisioning for Interdomain Quality of Service: the MESCAL Approach”, IEEE Communications Magazine, June 2005, pp. 129-137.

[101] MESCAL D1.2: “Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements”, January 2004, www.mescal.org.

[102] T.Ahmed, A. Asgari, A.Mehaoua, E. Borcoci, L.B Équille, K. Georgios “End-to-end quality of service provisioning through an integrated management system for multimedia content delivery” Computer Communications, Special Issue: Emerging Middleware for Next Generation Networks, Volume 30, Issue 3, 2 February 2007, Pages 638-651.

[103] M. Boucadair, P. Lévi, D. Griffin, N. Wang, M. Howarth, and G. Pavlou, E. Mykoniati and 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 Communication Magazine, Sept. 2007, pp. 134-143.

[104] Nikos Fotiou, et. al., “Developing Information Networking Further: From PSIRP to PURSUIT”.

[105] M. Rockl, K. Frank, P.G. Hermann, M. Vera, “Knowledge Representation and Inference in Context-Aware Computing Environments”, Proc. Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain, 2008.

[106] Open ContEnt Aware Networks (OCEAN), http://www.ict-ocean.eu/.

[107] TCPDUMP and LibPCAP public repository, available on-line: http://www.tcpdump.org/

[108] PCAP Express (PCAPx), “Gigabit Packet Capture acceleration,” www.npulsenetworks.com, 2008.

[109] L. Deri, “Improving Passive Packet Capture: Beyond Device Polling,” Proceedings of SANE 2004, 2004.

[110] L. Deri, "nCap: wire-speed packet capture and transmission," Workshop on End-to-End Monitoring Techniques and Services 2005, vol., no., pp. 47- 55, 15 May 2005, doi: 10.1109/E2EMON.2005.1564468

[111] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, D. Sadok, “A survey on Internet traffic identification,” Communications Surveys &Tutorials, IEEE, 11(3):37–52, August 2009

[112] Karagiannis T, Roido A, Aloutsos M, Laffy K. “Transport layer identification of P2P traffic”. In Proceedings of the2004 ACM SIGCOMM Internet Measurement Conference, ACM: New York, 2004:121–134.

[113] Internet Assigned Numbers Authority (IANA). http://www.iana.org/assignments/port-numbers, August 28,2010.

[114] Haffner P, Sen S, Spatscheck O, Wang D. “ACAS: Automated Construction of Application Signatures”. In SIGCOMM’05 Workshops, Philadelphia, PA, 2005: 197–202.

[115] S. Sen, O. Spatscheck, and D. Wang. “Accurate, scalable in-network identification of p2p traffic using application signatures”. In Proceedings of the 13th internationalconference on World Wide Web, ACM New York, NY, USA, 2004: 512-521.

[116] P.-C. Lin, Y.-D. Lin, T.-H. Lee, and Y.-C. Lai, “Using string matching for deep packet inspection,” Computer, vol. 41, no. 4, pp. 23–28, April 2008.

[117] Dharmapurikar., S., Krishnamurthy, P., Sproull, T. S., & Lockwood, J. W. (2004). Deep packet inspection using parallel Bloom filters. IEEE Micro, 52–61.

[118] T. Karagiannis, K. Papagiannaki, and M. Faloutsos.“BLINC: multilevel traffic classification in the dark”. In Proceedings on Applications, technologies, architectures, and protocols for computer communications, ACM New York, NY, USA, 2005: 229-240

[119] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, “Flow clustering using machine learning techniques,” in Proc. Passive and Active Measurement Workshop (PAM2004), Antibes Juan-les-Pins, France, April 2004.

[120] P. Cheeseman and J. Stutz, “Bayesian classification (AutoClass): Theory and results,” in Advances in Knowledge Discovery and Data Mining, 1996.

[121] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, “Traffic classification on the fly,” ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review, vol. 36, no. 2, 2006.

[122] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, “Semisupervised network traffic classification,” ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS) Performance Evaluation Review, vol. 35, no. 1, pp. 369–370, 2007.

[123] J. Garcia-Nieto, J. Toutouh, E. Alba, “Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics,” Engineering Applications of Artificial Intelligence, 23 (Issue 5) : 795 – 805, August 2010.

[124] S. F. L. Fernandes, C. Alberto Kamienski, J. Kelner, D. Mariz, D. Sadok, “A stratified traffic sampling methodology for seeing the big picture” Computer Networks, 52(14):2677–2689, 2008.

[125] T. Nguyen, G. Armitage, "A survey of techniques for internet traffic classification using machine learning," IEEE Communications Surveys, 2008

[126] A. W. Moore, K. Papagiannaki, "Toward the Accurate Identification of Network Applications," pages 41–54. 2005.

[127] RFC 3550: RTP: A Transport Protocol for Real-Time Applications

[128] RFC 5285: A General Mechanism for RTP Header Extensions

[129] TCPDUMP and LibPCAP, on-line: http://www.tcpdump.org, accessed: May 2013

[130] IPTABLES, available on-line: http://www.netfilter.org/projects/iptables/index.html

[131] IPROUTE2, available on-line: http://www.linuxfoundation.org/ collaborate/ workgroups/networking/iproute2

[132] Linux Traffic Control, available on-line: http://linux-ip.net/articles/Traffic-Control-HOWTO

[133] MPLS Linux Project. available on-line: http://sourceforge.net/apps/mediawiki/mpls-linux/index.php?title=Main_Page

[134] A. Dainotti, A. Botta, A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios", Computer Networks (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547

[135] Luca Deri, Netikos, Via, Loc La Figuretta, "Improving passive packet capture: beyond device polling", In Proceedings of SANE 2004, 2004

[136] Howarth, M.P. et al., "Provisioning for Interdomain Quality of Service: the MESCAL Approach", IEEE Communications Magazine, June 2005, pp. 129-137.

[137] E.Borcoci (ed) et.al., ICT ALICANTE, Deliverable D6.2.1: CAN Management, Control and Interfaces –I, Feb. 2012, http://www.ict-alicante.eu/.

[138] J. Li, Y. Zhang (eds.) et al., ICT ALICANTE, Deliverable D5.1.1: "Service Provider Environment – I", September 2011, http://www.ict-alicante.eu/

[139] T.Ahmed, A. Asgari, A.Mehaoua, E. Borcoci, L.B Équille, K. Georgios "End-to-end quality of service provisioning through an integrated management system for multimedia content delivery" Computer Communications, Special Issue: Emerging Middleware for Next Generation Networks, Volume 30, Issue 3, 2 February 2007, Pages 638-651.

[140] G. Gardikis (ed.) et al., ICT ALICANTE, Deliverable D6.5.1: "CAN Layer Monitoring and Security", September 2011, http://www.ict-alicante.eu/.

[141] Internet2, OWAMP version 3.3. http://www.internet2.edu/performance/owamp/

[142] T. Auld, A. W. Moore, S. F. Gull, "Bayesian neural networks for internet traffic classification," Neural Networks, IEEE Transactions on, 18(1):223–239, 2007.

[143] R. Banner and A. Orda, "Multipath routing algorithms for congestion minimization",Proc IFIP Networking, 2005.

[144] I. Cidon, et al., Analysis of Multi-Path Routing, IEEE/ACM Trans. on Networking, 7(6), pp.885-896,1999.

[145] Honma Y, Aida M, Shimonishi H, Iwata A, A New Multi-Path Routing Methodology Based on Logit Type Assignment, GLOBECOM Workshops, 2009 IEEE.

[146] P. Narvaez and K. Y. Siu, Efficient Algorithms for Multi-Path Link State Routing, In Proceedings of ISCOM'99,1999.

[147] C. Hendrick, Routing Information Protocol, RFC 1058, 1988.

[148] J. Moy, OSPF version 2, RFC2328, 1988.

[149] P Anapliotis, E Markakis, A Sideris, E Pallis (TEIC), D.Negru (LABRI), IC, “A novel content-aware multipath routing concept exploiting random utility theory principles” in Proc. International Conference on Telecommunications and Multimedia (TEMU2012), Heraklion, Crete, Greece, 30 July-1 August, 2012.

[150] Steven S. Skiena, “The Algorithm Design Manual”, Second Edition, Springer, ISBN: 978-1-84800-069-8.

[151] G. Xilouris, G. Gardikis (eds.) et al., ICT ALICANTE, Deliverable D2.1: "ALICANTE Overall System and Components Definition and Specification", September 2011, http://www.ict-alicante.eu/.

[152] M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D2.3: "Cross-layer Monitoring Definition and Specification", September 2011, http://www.ict-alicante.eu/.

[153] R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D2.4: "Definition and Specification of the ALICANTE Pilot Architecture", September 2011, http://www.ict-alicante.eu/.

[154] M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D4.1.1: "The ALICANTE Home-Box Layer – I", September 2011, http://www.ict-alicante.eu/.R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D4.2.1: "Home-Box Layer Interfaces and Monitoring – I", September 2012, http://www.ict-alicante.eu/.

[155] E. Markakis, (ed) et al., ICT ALICANTE, Deliverable D4.3.1: " Home-Box Layer Services – I", September 2012, http://www.ict-alicante.eu/.

[156] M. Sidibe, (ed) et al., ICT ALICANTE, Deliverable D4.4: " The ALICANTE Home-Box Layer – Final", May 2013, http://www.ict-alicante.eu/.

[157] A.Mevel (ed) et.al, ICT ALICANTE, Deliverable D6.1.1: “Content-Aware Network Infrastructure and Elements – Intermediate”, September 2011, http://www.ict-alicante.eu/.

[158] E.Pallis (ed.) et al., ICT ALICANTE D6.3.1: “Network Layer Management and Control”, July 2012, http://www.ict-alicante.eu/

[159] E.Borcoci (ed.) et. al. ICT ALICANTE D6.6: “The ALICANTE CAN and Network Environment”, June 2013, http://www.ict-alicante.eu/.

[160] R. Salgado, (ed) et al., ICT ALICANTE, Deliverable D8.2: " The ALICANTE Pilot Integration – Final", September 2013, http://www.ict-alicante.eu/.

[161] Y. Zhang (ed) et al., ICT ALICANTE, Deliverable D8.3: "Trials and Validation", September 2013, http://www.ict-alicante.eu/.