Ph.D. in Electronic Engineering and Computer Science

Dept.of Electrical and Electronic Engineering

University of Cagliari, Sardinia, Italy

Architectures and Technologies for Quality of Service Provisioning in Next Generation Networks

Ing. Giovanni Branca

Advisor: Dott.Ing. Luigi Atzori

Curriculum: ING-INF/03 – Ph.D. in Electronic Engineering and Computer Science

XXIV Cycle

March 2012

To my parents
and
important people of my life

Contents

Introduction 5

Related papers 9

Chapter 1. Next Generation Networks 10

1.1 Definitions and architectural models 10

1.2 Quality of service (QoS) 14

1.3 Technologies for QoS in packetized networks 16

Chapter 2. Service Oriented Architectures in Next Generation Network Architectures 20

2.1 The Service-Oriented Architecture paradigm 20

2.2 SOA in telecomunications 22

2.2.1 The Web Services technologies and SOAP protocol 23

2.3 SOA in Next Generation Networks 26

2.3.1 SOA in the Service Stratum 27

2.3.2 SOA in the Transport Stratum 40

2.3.3 SOA in Management Technologies 45

2.4 A proposal to use SOA in the Transport Stratum 49

2.4.1 Transport stratum architectural model 50

2.4.2 System evaluation 59

2.4.3 Experiences in development and deployment and conclusions 63

Chapter 3. Bandwidth Management in Next Generation Networks 64

3.1 Introduction 64

3.2 Autonomic Traffic Engineering 65

3.2.1 Traffic Engineering functionalities 65

3.2.2 Autonomic Approach 66

3.3 Bandwidth Management in DS-TE 67

3.4 Distributed Bandwidth Self-Management 68

3.4.1 The basic principle 70

3.4.2 The algorithm 71

3.5 Experimental Results 75

3.6 Conclusions 77

Chapter 4. Peer to Peer Video on Demand Network Architectures 79

4.1 Introduction 79

4.2 The P2P-VoD Scenario 81

4.3 Models 84

4.4 Simulations 90

4.5 Conclusions 94

Chapter 5. Conclusions 95

Acknoledgements 98

References 99

Introduction

Today, Next Generation Networks are attracting many interests of telecommunication players. Unlike the initial market specialization, where a player was often clearly identified with the service provided, the trend followed in recent years has transformed the players in multi-players. Many services are now provided by a single player, so the customer is attracted by combination of services offered rather than the single service characteristics. In addition, the strong competition make the services price more and more decreasing and at the same time new services are proposed to customers. These evidences introduce an hard competition between players that are driving their strategies towards the customer satisfaction, but also have the desiderata to reduce costs and rationalize information and communication delivery architecture.

In this context, Next Generation Networks are expected to fulfill the custom-need-driven model, which is mainly intended to: satisfy customers in differentiated ways, quickly deploy new services at user request, manage network and service resources in an integrate way. Accordingly, the ITU-T NGN architecture is characterized by the separation of the service stratum from the transport stratum so as to enable: the flexibility to add, maintain and remove services without any impact on the transport layer; the flexibility to add, maintain and remove transport technologies without any impact on the access to service, application, content and information; and finally the optimized usage of multiple access and core transport technologies to form end-to-end connectivity across multiple terminals, different access technologies and different core transport technologies. The service stratum is then responsible for enabling the creation and the delivery of service, application, content and information. As to the transport stratum, it is intended to provide end-to-end connectivity according to the service requirements differentiating the service treatment, the terminal capability and status of the network resource availability.

Differentiation of services in terms of Quality of Service (QoS) is then a must for the transport stratum, which is now possible thanks to the technological advancements achieved in the decade that allow for differentiating traffic delivery in the access and the backbone networks. In a similar way, other works have brought to protocols and technologies purpose-built to support Traffic Engineering operations (TE): they permit to have the optimal traffic distribution avoiding bottlenecks, with consequent link congestion and data loss. The union of TE and traffic delivery differentiation has brought to the development of adequate tools for QoS management, so that network operators have a more efficient alternative than the classic (and more expensive) network over-provisioning. Technology is then not an issue when differentiating services at flow or trunk granularity as well as when engineering the network resources and traffic.

However, a big challenge remains in this context, which is related to the flexibility and dynamicity in the provisioning of new transport services when requested by the service stratum so as to follow the rapid evolution of the user needs and market trends, which is one of the main features of the NGNs. Herein, dynamicity is intended as the capability of the network to bind on-demand services and relevant resources, often provided by different operators through separate domains, at user request and according to his profile. Indeed, the creation of new market-driven applications by reusing an extensible set of existing service components has been a key aspect of telecom platforms for years, but this has almost exclusively characterized the application/service stratum. In truth, application provisioning is currently carried out without the cooperation of the lower layers in the transport stratum, which are responsible for the delivery of the data between distributed service entry-points. Contrarily, to meet the previously mentioned challenge of dynamically providing on-demand services, there is the need for transport functionalities to respond to the following demand: provide the upper layers with interoperable and open interfaces so that dynamic binding of distributed application services is synchronized with dynamic activation, configuration and monitoring of transport services.

This thesis goes in this direction, with three topics related to NGN network architectures and services that requires QoS guarantees. The first topic studies the potentialities to make available in a flexible and dynamic way all the capabilities of the transport stratum to the upper layers (i.e. the service stratum), to the end-user functions and to other networks. The addressed challenge is the analysis of the transport functional layer as required in the NGN transport stratum, the decomposition of this layer into a set of micro-functionalities according to the Service Oriented Architecture (SOA) style and the implementation of the relevant services using the Web Service (WS) technologies to analyze the advantages of the proposed approach. In particular, we considered the DiffServ-aware Traffic Engineering (DS-TE) solution, which represents one of the most advanced technologies that performs TE in a differentiated service environment by applying routing constrains with class granularity.

In the second topic, we focus on specific QoS technology, the DiffServ-aware Traffic Engineering (DS-TE) architecture. It implements TE in a differentiated service environment by introducing the concept of bandwidth constraint model (BC Model). A BC Model defines the percentage of resources which are allocable in each link to each traffic class type to grant QoS requirements and traffic load balancing and are used by the CBR algorithms to perform more accurate routing decisions based on the local specific class-based rules on bandwidth utilization. These constraint-based routing (CBR) algorithms work based on network state information and require to be supported by specific tools of traffic demand estimation, QoS policy definition, network performance monitoring and bandwidth management, which specify resource availability and other constraints for the optimal accommodation of the traffic flows. Due to their complexity, these tools are often deployed to work on a centralized TE unit, which collects all network information in a central data repository and remotely configures some TE parameters in the network nodes. At the same time, CBR algorithms are implemented by edge routers, which manage all control plane functionalities on the network to compute the optimum end-to-end path for each packet, while the core nodes are often limited to switching operations high forwarding rate performance. Although this approach grants high forwarding rate performance on the core network, it is often not able to efficiently perform network configuration. Indeed, TE network state evolves dynamically in the amount of resources available due to the continuous optimization actions, setting up of new paths and preemption, rerouting or termination of the existing ones so that state information on the central data repository may result out-of-date. Hence, with the increasing of network complexity, the network management is evolving towards autonomic approaches. Autonomic management allows network nodes to automatically discover their environment, self-configure and automatically update to adapt to changes. The objective of traffic self-optimization is to evaluate all or part of the possible paths that satisfy the requirement of the traffic requests and select the best one to route the flow. When implementing an autonomic approach for traffic optimization, without human intervention, the network autonomously computes the best paths so as they evolve to a stable and robust control point.

In this scenario, we present a distributed bandwidth management approach, which provides for a self-management module in each node of the network to improve the efficiency of TE functionalities while limiting control plane complexity at core nodes. We propose an implementation of the algorithm for two standard BC Models, the Maximum Allocation Model (MAM) and the Russian Doll Model (RDM), but it can be easily adapted to other standard or non-standard models. The effectiveness of the proposed solution is evaluated with respect to traffic distribution, bandwidth blocking rate and preemption rate parameters.

Finally, in the context of the NGN network architectures and services, we have addressed the hot problem of the video transmission, specifically the Video On Demand service delivery in peer-to-peer based networks. According to several traffic analyses, video transmission applications over the Internet seems to be the one application will drive the Internet evolution during the near future. One interesting application is the Video on Demand service, which requires resources able to deliver a video whenever the customer request it. Realizing a VoD system using the Internet requires architectures tailored to video characteristics. Peer-to-peer networks architectures (P2P) are becoming more and more popular in video content delivery services, such as TV broadcast and Video on Demand (VoD), thanks to their scalability feature. Such characteristic allows for higher numbers of simultaneous users at a given server load and bandwidth with respect to alternative solutions. However, great efforts are still required to study and design reliable and QoS-guaranteed solutions.

Within the scenario of P2P-based VoD services, we study the phenomenon of peer churns and propose four models of the peer behaviour to evaluate its impact on the system performance, which are based on: the Gilbert-Elliot chain, the fluidic representation of the user behavior and a queuing analysis of the system. The models are compared by computing the resources the system has to add on top of the P2P network to satisfy all the download requests; they provide an important tool for service providers to evaluate resources to add on the system architecture with the aim to guarantee an adequate level of quality of service for final users.

Related papers

- **Giovanni Branca**, Luigi Atzori "A Survey of SOA Technologies in NGN Network Architectures" *IEEE Communications Surveys and Tutorials,* 2011, to appear.

- **Giovanni Branca**, Paolo Anedda, Luigi Atzori "Transport Stratum Services in NGN: a SOA-oriented design" *IEEE Global Communication Conference 2010 (Globecom 2010),* Miami, Florida, USA, Dec.2010.

- Tatiana Onali, **Giovanni Branca,** Luigi Atzori "Bandwidth Self-Management in NGN Networks" Telecommunications Networks, InTech - Open Access Publisher, 2011, under review.

- Luigi Atzori, Tatiana Onali, **Giovanni Branca** "Bandwidth Self-Management in DS-TE Networks" *IEEE Global Communication Conference 2011 (Globecom 2011)*, Houston, Texas, USA, Dec.2011.

- **Giovanni Branca**, Thomas Schierl, Luigi Atzori "Theoretical Models for Video on Demand Services on Peer-to-Peer Networks" *International Journal of Digital Multimedia Broadcasting - Special Issue on Convergence of Digital TV Systems and Services*, Hindawi Publishing Corporation, July 2009 (doi:10.1155/2009/263936).

- **Giovanni Branca**, Thomas Schierl, Luigi Atzori "Evaluating Peer Churn Effects on P2P-based Video-on-Demand Services, *IEEE Global Communication Conference 2009 (Globecom 2009),* Honolulu, Hawaii, USA, Dec.2009.

Chapter 1. Next Generation Networks

1.1 Definitions and architectural models

Since Internet was born, we have experienced its expansion regarding both the number of users and the number of different services available. As a consequence of this rapid expansion until today, service providers have more and more the needs to speed up the implementation of new network solutions in a effective and efficient way. These newest and innovative network solutions are generally referred to as Next Generation Networks (NGN). The overfullness of experimental solutions, forum industries, customized solutions and academical proposals regarding NGNs can create some kind of confusion, so we think that the best approach toward the study and the research in NGN is to begin with standards definitions. To avoid misunderstandings, we refer to standard definitions that have been accepted by all the industries and research community.

The main international organizations that are working towards the standardization of new telecommunication networks and the definition of relevant guidelines are the ITU-T [1] and the ETSI-Telecommunication and Internet converged Services and Protocols for Advanced Networking (TISPAN) standardization body [2]. According to International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), the definition of NGN is:

"a Next Generation Network is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, Quality of Service-enabled transport technologies and in which service–related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users."

The ITU-T definition highlights three main important features:

- no specific technologies are selected and characterize the NGN networks.
- QoS functionalities are crucial in this architecture.
- service-related functions are independent from underlaying transport-related technologies.

The typical constituents in NGN can be illustrated in Figure 1.1, where networks are classified in access and core networks. The access networks involve disparate wireless and wireline access technologies to provide ubiquitous services to end users, as can be seen in the external circle in the Figure 1.1. The core networks take care to carry the data across the network: they include two major technologies that are the legacy technologies such as Asyncronous Transport Mode (ATM) and the modern family of IP-based cores. IP-based technologies such as Multi-Protocol Label Switching (MPLS) possess two QoS models standardized by IETF, the Diffentiated Services and the Integrated Services model.

The end-to-end communications can embrace mobile, wireless and fixed networks of several operators and multiple network technologies. NGN is expected to employ multiple networking technologies for the best service provisioning; the presence of different actors and disparate transport technologies has the effect to define a complex scenario with important challenges for interconnection, interworking and interoperation between network technologies and telecommunication operators.

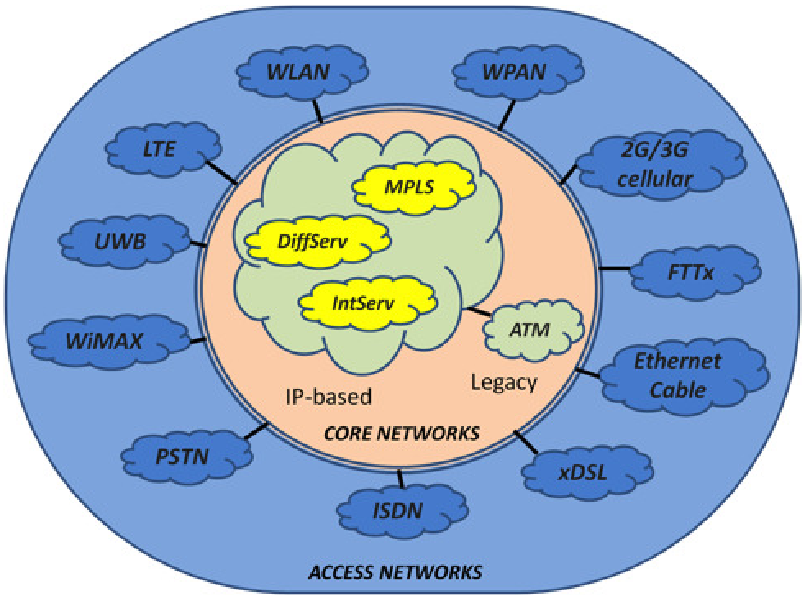

Figure 1.1. Typical constituents of Next Generation Networks grouped in access and core networks technologies.

With the aim to create a logical framework for NGN, the main standardization bodies also propose in the ITU-T Rec Y.2001 [1] a functional model, which also corresponds to what has been proposed by ETSI in 2009 [04]. As shown in Figure 1.2, the functional model is mainly organized in three different layers: the service stratum, the transport stratum and the application layer.

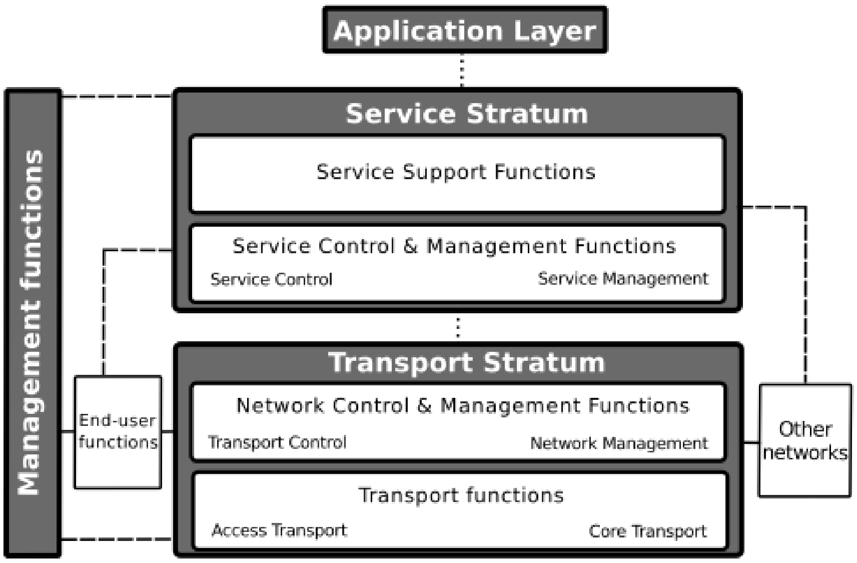

Figure 1.2 - Next Generation Networks functional architecture proposed by ITU-T

The transport stratum takes care of maintaining the end-to-end connectivity according to the service requirements, terminal and resource capability. It means that the network should be able to take care of the traffic flows according to the quality of service requirements, which depend on the service provided (e.g.: voice, video streaming, web browsing, email download) and to the user terminal (e.g., user accessing the network through a smartphone, a desktop or a laptop). Other than data transport functionalities at the core and access components of the network, this stratum includes a sublayer for the control and management of these functionalities. Such a sublayer is devoted to the management of the end-to-end connectivity as well as to the inventory, configuration and performance monitoring.

The transport stratum then provides appropriate interfaces to the service stratum that can: obtain information about the transport capability, setup an end-to-end connection over the network with certain QoS requirements, have information about the status of active connections, close an active connection, change the egress node for active connections. Accordingly, the service stratum is responsible for enabling the creation and the delivery of services and applications. The main foreseen functions provided by this stratum are: user authentication, call setup, managing group of users, managing the user profile, messaging, presence availability info, streaming services, video services, mobility management, cloud services, and others. Indeed, these functionalities are provided by a service control and management functions sublayer, whereas the service support functions sublayer is responsible for providing the enablers for the creation of service, application, content and information to the internal or third-party providers. All these enablers can be invoked with open APIs (Application Programming Interface).

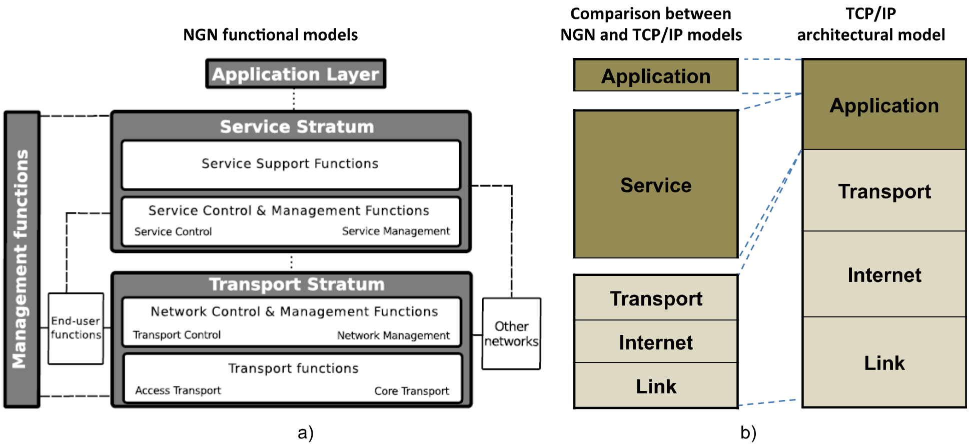

Figure 1.3 - Architectural models: a) ITU-T NGN functional model and b) Comparison between ITU-T NGN model and classical TCP/IP model

Although the transport and the service strata have internal management functions, in the architecture the more generic management functions are located in a separate block, which has interfaces with the transport and service strata and with the end-user functions block.

The application layer is composed of servers that can use various sets of APIs to discover and to use the available enablers and finally to create and execute their own service, application or content.

The reduced number of network layers represents the industry efforts in reducing network overhead and system complexity and it is also one of the most significant updates by the NGN. Additionally, it has to be noted that the NGN standards released by ITU-T just define an overall framework with its requirements, but no mandatory network technologies have been defined. This situation encourages healthy competition in the telecommunication market by enabling network solution providers to use their preferred technologies to fulfill the envisioned NGN framework once the final solution meets the NGN requirements.

This approach is quite different from the IP model, which is based on a higher number of layers, an exact suite of core protocols to be used, and a limited set of functionalities when compared to the NGN. The latter indeed adds functionalities addressing issues mainly related to: interoperability, multi-domain management, network service management, and traffic differentiation. In Figure 1.3, we provide a comparison between the old TCP/IP architectural and the NGN functional models. It can be noted that in the original Internet architecture great consideration was given to the access and internetworking issues, which are addressed in the first three layers. These have now lost importance leaving room for the service layer. Indeed, the service stratum has acquired part of the old application layer and many new functionalities have been added, which originally were not crucial for service provisioning and management due to the narrower set of application scenarios with respect to current and future ones. The traditional model is a hierarchical structure where each service is statically defined and invoked, and cannot be recombined with others. The service interfaces are the result of a long standardization process: the N-th layer can communicate only with the closest layers, i.e. with the N+1 and N-1 layers. It is clear that this model can not perform the request of NGN, where there is the need for a more flexible environme nt for the dynamic creation of services.

In the late ’90, the TINA consortium [4] has proposed a similar approach based on service-based network architectures: in its guidelines, it introduces the concept that a network architecture should be designed according to a service-oriented abstraction of network resources. For example, the network architecture has to provide a service-oriented view procedures related to the connectivity, such as creation and management of connections. In a TINA architecture, any use of the network is the result of using a service, and for this reason it is important that the network provides a service-oriented view of connectivity procedures. In this thesis, TINA principles are an important starting point for the problem analyzed.

1.2 Quality of service (QoS)

The basic idea to concentrate all existent and new services delivered to the final user in a unique network is a big challenge. The requirements of a better management with less resources and at the same time the better customer satisfaction lead to the deployment of multiservice networks with Quality of Services (QoS) issues.

Commonly, the Quality of Service (QoS) term refers as "a defined measure of performance in a data communication system". With reference to modern telecommunication networks, it is the mechanism in the network that set a priority for delivering packets. QoS key parameters and their measurements are based on well-defined characteristics of the applications considered. In turn, the QoS level measured corresponds in the satisfaction of the final user, which is usually measured with the Quality of Experience (QoE) level. QoS key parameters are strictly related to technical measurement over the network whereas QoE parameters are usually based on a subjective measures that relies on human opinion. QoE parameters and their uses are out of the scope of this thesis.

Basically, given an application in a packet-based network, i.e. an IP network, the main QoS key parameters that impacts in the services are the following:

- delay: this parameter includes the time separating the user request action and the establishement and delivery of the service requested. Delay has a big impact on the user experience, with various aspects that depend on the application considered. Usually it is the sum of delays introduced by the network, the user terminal and the computing time.

- delay variation (jitter): it measures the difference between the delay of each data-packet in a delivered stream. It impacts in the delay intolerant applications, with the degradation of the quality perceived.
- information loss: the amount of information not arrived to destination has the effect to increase the bit error rate and consequently to reduce the quality of the voice, video, image or data decoded and experienced by the end user.

For example, if we consider some basilar services such as audio and video transmission, the final user satisfaction depends on different combination of the key parameters above described. In Table I just few typical requirements are shown; many other parameters have been defined but are not cited here for the sake of simplicity.

Table I : Impact of bad QoS key parameters in typical applications

	Delay	Jitter	Information loss
Conversational voice	High	High	Medium
Video streaming	Low (using buffering)	Low (using buffering)	High
Videoconference	High	Medium	High
Video on demand	Medium (depends on buffering)	Low	Low
Data transmission	Low	Low	Low

Actually, the classical IP network protocol is not able to guarantee different QoS level requirements. The IP protocol considers packet transmitted with no differentiation because it works according to the best-effort policy. The data (and the correspondent services) are transmitted with no guarantees on packet delivery, ordering and prioritization; routing decisions are taken hop-by-hop just considering the IP destination address: the simplicity of this protocol has determined its success respect to other network protocols. Anyway, the best-effort approach is not effective with the new applications that have rigorous quality of service requirements, especially when different data traffic is transmitted in the same network links. The most simple solution to overcome these limits could be the overprovisioning of the network capabilities, for example the bandwidth increasing, but this solution is clearly not efficient. To overcome this limit, many efforts have been done to provide an adequate QoS level in IP- based networks.

1.3 Technologies for QoS in packetized networks

In the last years, the most important technologies developed for the QoS are the Integrated Service (IntServ), the Differentiated Service (DiffServ), the Multi Protocol Label Switching (MPLS), the Traffic Engineering (TE) and finally the Differentiated Services - Traffic Engineering (DS-TE) technology. Actually, the last one represents the state-of-art solution in packetized networks, more specifically for IP-based networks. In this section, these technologies are briefly explained.

- ***Integrated Services***

The Integrated Services (IntServ) [5] is the first attempt to provide quality of service in IP-based networks. Basically, the IntServ technology introduces a connection-oriented model with the aim to associate a single connection end-to-end per each data flow. Flows maintain the same properties all over the network path, for each hop along the network nodes. IntServ is based on the Reservation Protocol (RSVP); for each incoming flow, it must be created the path over the network nodes and only after this step the data can be sent into the network.

This approach has the big problem to be unscalable and unreasonably complex. Notwithstanding many attempts to use it, it was an unsuccessfully technology.

- ***Differentiated Services***

DiffServ technology [6] was born to overcoming the limits of IntServ, expecially the scalability problems. DiffServ technology adopts a class-based approch to differentiate data (and, in turn, services) in the network. The incoming traffic requests are classified and aggregated into few Class-of-Service (CoS) groups which correspond to precise quality requirements. Every packet is marked with a 6-bit field, which is incapsulated into type of service (ToS) field of IP packet header. This field is called Differentiated Service Code Point (DSCP).

DiffServ is connection-less technology, and the routing decisions are totally in charge of classical routing protocols. The DSCP defines the per-hop behaviour (PHB) of the packet regarding the scheduling and the drop precedence. In a DiffServ domain there can be up to 64 PHB, but, in practice, only few CoS are commonly used:

- Best Effort (BE): not differentiated treatment
- Expedited Forwarding (EF): minimal delay and low loss. Voice and video flows belong to this class, due to little delay and low packet loss requirements.

- Assured forwarding (AF): packets populate different queues with the same characteristic for the forwarding. Separated queues are ideal for traffic that require rate assurance but not constraints on the delay or jitter.

In addition to the traffic classification, edge routers also execute traffic conditioning functions:

- Shaping: traffic exceeding rate defined by its profile is buffered and transmitted when it can respect the same profile rules: the aim is to avoid congestion.
- Policing: the traffic is classified into aggregates, that are processed in order to respect more general service level agreements (SLA).
- Metering: traffic is monitored to guarantee the compliance with the traffic profile defined. When irregularities are found, metering functions communicate with other functions for the traffic re-classification or dropping.

DiffServ technology overcomes the IntServ problems of scalability, but it shows the limit to work at network level in the TCP/IP stack. Each packet needs to be analyzed until the network level, and this behaviour introduces more delay in terms of processing time.

o *Multi Protocol Label Switching*

Another solution for the QoS in IP-based networks is the Multi Protocol Label Switching (MPLS) [7]. The MPLS technology is a flow-based protocol with an evolutionary approach with respect to Asyncronous Transfer Mode (ATM) and Frame Relay (FR) connection-oriented legacy technologies. These technologies use the concept of virtual circuit to set up a path before start data transmission over the network.

MPLS deploys the same idea of connection-oriented technologies in the IP networks with the implementation of the label switching paradigm. The Label switching technology is aimed to create a path from the ingress to the egress point by distributing a set of label correspondences. In a MPLS domain, the path is called Label Switch Path (LSP) and the routers are the Label Switch Router (LSR). MPLS adds an header between the layer 2 (logical connection) and the layer 3 (network information), so that each router in the MPLS domain has the only task to read the MPLS header, find the next hop and replace the old label with the corresponding new label (label swapping). For this reason, MPLS works at 2.5 level in the TCP/IP stack, integrating IP technologies with the advantages and the velocity of switching technologies at layer 2.

In a MPLS domain, the edge routers E-LSR are able to work both IP routing and label switching and they represent the interface with external domains; they also take care to classify the incoming traffic in a Forwarding Equivalence Class (FEC) and set up the packet with the first label corresponding to a specific LSP.

- ***Traffic Engineering***

Traffic engineering addresses the problem to efficiently allocate resources in the network so that user constraints are met and the operator benefit is maximized. IP routing protocols calculate the best path from each point of the network and the egress point. The key information is the IP destination address, which is contained in each packet header. This feature has the effect to route all the traffic only in the shortest paths, with the creation of congestions in these paths and consequently the starvation of other links.

Traffic engineering techniques introduce more complex routing protocols that consider also other parameters such as the bandwidth avaliable or the delay introduced by each link. TE routing protocol class is known in literature as Constraint Based Routing (CBR) protocols. These constraint-based routing (CBR) algorithms work based on network state information and require to be supported by traffic demand estimation, QoS policy definition, network performance monitoring and bandwidth management, which specify resource availability and other constraints for the optimal accommodation of the traffic flows

MPLS technology is used to realize the traffic engineering using the traffic trunk (TT) concept. A TT is a traffic aggregation of flows which belong to the same FEC and are sent in the same LSP. The path is computed by CBR according to TE requirements, the traffic is aggregate in a TT and sent over the corresponding LSP. LSP are characterized also by priority and preemption attributes. The former defines the importance of the LSP with respect to others. The latter defines the availability to preempt another LSP in a given path if there is competition for available (and limited) resources.

- ***Differentiated-Services with Traffic Engineering***

Actually, DS-TE technology [8] is considered as the state-of-art solution to provide and manage quality of service features in IP-based networks. It combines the advantages of DiffServ and MPLS-TE networks; in a DS-TE domain, the Traffic Engineering is implemented in a per-class basis by defining the concepts of class type (CT), TE-class and bandwidth constraint (BC) model. A CT is a set of traffic trunks with similar QoS requirements and the same constraints on bandwidth utilization. A TE-class is a combination of a CT and a preemption value. It allows traffic flows belonging to the same CT to be forwarded over different path at different priorities and to preempt bandwidth in use by paths with lower preemption priority. Finally, a BC model defines the amount of links bandwidth that can be used to route traffic belonging to each CT. It allows the CBR algorithm to determine the amount of available resources and the possibility to set up the new LSP in the available paths to the destination, evaluating pre-emption actions on old LSPs.

Currently, the IETF proposes two main standard BC Models: the MAM and the RDM. According to the MAM, the link bandwidth is divided among the different CTs, defining a different constraint for each one. This model achieves complete isolation between different CTs, so that it is not necessary to define priorities between LSPs carrying traffic from different CTs. The problem of MAM is the risk of wasted bandwidth, so that the RDM was proposed to allow for bandwidth sharing among CTs with the constraint that a CT can preempt the bandwidth to a lower priority CT. In particular, if M is the number of active CTs, RDM defines M BCs with the m-th representing the maximum bandwidth allocable for CTm and that can be shared among all CTs between CT0 (the lowest priority one) and CTm. A BC is defined for each CT with the means that the BC for the z-th CT represents the maximum bandwidth which can be shared among all CTs with priority lower or equal to the z-th CT. In addition to the IETF models, other bandwidth allocation schemes have been defined for DS-TE to address uncommon and complex application scenarios.

Chapter 2. Service Oriented Architectures in Next Generation Network Architectures

2.1 The Service-Oriented Architecture paradigm

For the Information and Communication Technology (ICT) researchers and workers, the term Service-Oriented is commonly referred to as a generic theory that can be applied to solve a variety of problems. This theory follows the simple logic for that a complex and large problem can be solved in a shorter time and with less resources if it is decomposed into a collection of simple, small, single-function parts. Each of these parts deal with a specific part of the problem or a concern and according to the service oriented approach, each part represents a "service". A composition of services for building a complex system represents a Service-Oriented Architecture (SOA).

According to the Organization for the Advancement of Structured Information Standards (OASIS) [9] a SOA is "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains". It provides a uniform means to offer, discover, interact with and use capabilities provided by different systems; the aim is to increase the agility, efficiency and effectiveness during systems development and integration.

Notwithstanding the SOA essence is delineated by the OASIS reference model, the SOA abstraction is not constrained to be a specific technology, but only defines some properties oriented to the reuse of single services and to the integration in an heterogeneous environment [10]. SOA and service-orientation approach are implementation-agnostic paradigms that can be realized with any suitable technology platform. In the same OASIS definition the standards are implicit: the use of industry standard interfaces and the creation of services without dependencies implies more flexibility to realize the service composition and to partner dynamically with new services and complex systems.

If we think to condense the SOA essence in only two key words, the first one is certainly the word "service". As a matter of fact, a service-oriented architecture is composed by many small parts called services. In SOA, a service can be defined as an "application function packaged as a reusable component for use in a business process"[11]. According to this definition, a SOA-based service must be designed and developed following specific architectural principles:

- Encapsulation: each service has well-defined capabilities and is able to communicate with other services with standardized interfaces;
- Loose coupling: services maintain a relationship with others that minimize dependencies and only require they maintain an awareness of each other;
- Contract: services subscribe to a communication agreement with other services;
- Abstraction: services hide their own internal logic from the outside world;
- Autonomy: services have to be well-defined, complete and independent from the context or the state of other services;
- Reusability: the logic is divided into services with the intention of promoting reuse;
- Composability: many services can be coordinated and assembled to form complex services;
- Stalelessness: services minimize retaining information specific to an activity;
- Discoverability: services are defined to be externally descriptive so that they can be found and assessed via an available discovery mechanism.

Another key aspect should not be neglected: the services strongly depend on their interfaces, which publish the internal functionalities of a service. If a service does not expose in a correct and complete way the interfaces with its own functionalities or characteristics, it corresponds to hiding its capabilities. [12]. In essence, services within the SOA are self-contained, modular, interoperable, loosely coupled, location-transparent, composite entities that can share formal contracts with other services in the shape of service level agreements (SLAs).

The second key word in the SOA is “composition”. After the elementary services have been defined following the principles explained in the previous section, it is possible to build complex and final services for the end-user business processes and applications. A combination of more than one service leads to the final result. The process of composition is organized by the publish, find, bind and invoke paradigm. In the Figure 2.1 it is illustrated the correspondent framework with its components.

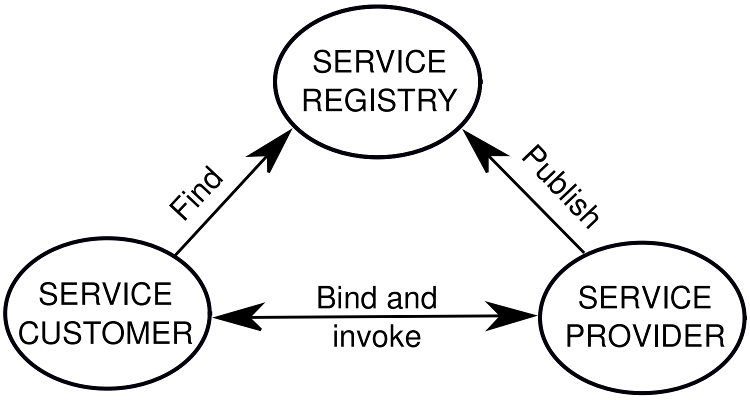

Figure 2.1. SOA collaborations: publish, find, bind and invoke paradigm.

In this process, three types of components can be recognized: the Service Provider, the Service Consumer and the Service Registry. The Service Provider is the component that makes a service available. Subsequently, the service has to be published to become available to other services. The service provider sends to the Service Registry its own useful information to be published. Accordingly, the Service Registry maintains some information about every service how the service can be invoked and all functionalities that the service is able to provide. When a Service Consumer wishes to use a service, it gets relevant information in the Service Registry, that is used to know how to communicate with the Service Provider. Hence, the SOA framework is organized according to publish, find, bind and invoke paradigm. The use of this paradigm introduces a higher level of flexibility in the creation of complex services, built with an interconnection of many elementary services: it is possible to create in a transparent way many services for the end-user in a short time, using the composition of elementary services.

Summarizing, a SOA-based architecture provides a loosely-integrated suite of services that can be used within multiple business domains. The SOA abstraction is not constrained to a specific technology, but only defines some properties oriented to the reuse and to the integration in an heterogeneous environment.

2.2 SOA in telecomunications

Initially, in the field of Information and Communication Technologies (ICT) the SOA principles have been firstly adopted in the software design. Extending the principles explained in the previous section to the software, a service is a software component and the composition rules are represented by the software architecture. The former matches the characteristics listed above, whereas the latters defines how the components interact with each other.

The ideas of SOA are easy to implement in the software field. At the contrary, the adoption of the SOA ideas to telecommunications and more specifically to network architectures is more complex, so that standardization efforts are required. A telecommunication network is made of physically distributed, complex and heterogeneous systems that often are under different ownership domains. Hence, the design or re-design of each single service to be compliance with SOA principles is not a simple and immediate task.

In the last decade, SOA principles have been adopted by the telecommunication research community and industry, promoted by the need for an open set of reusable components to support the efficient design, deployment, provisioning and management of seamless services for various business models across different networks. Indeed, the SOA technologies contribute to the creation of scalable service infrastructures by stimulating the evolution of existing services, cutting costs, leading the innovation and providing the Quality of Service (QoS) requested by the end user with the appropriate network resources.

2.2.1 The Web Services technologies and SOAP protocol

A Web Service (WS) can be defined as a software application accessible through an URL using XML-based protocols, such as Simple Object Access Protocol (SOAP) messages sent over Internet protocols. Recently, many efforts have been done towards WS standardization: following the recommendations of these standards, a WS can be an effective realization of a service according to SOA style. In fact, implementors commonly build SOAs using Web Services standards that have gained broad industry acceptance. These standards also provide greater interoperability and some protection from closed proprietary vendor software.

The SOAP is the transport protocol for messages sent and received by the Web Services. It exchanges XML-based messages in a decentralized and distributed environment, such as a telecommunication network, using the HTTP (Hypertext Transfer Protocol) protocol. A SOAP message consists of three parts: the first one is the envelope, which defines the framework for describing what a message contains and how to process it; the second is a set of encoding rules and the third is a convention for representing Remote Procedure Calls (RPC) and the responses.

In the last few years a lot of effort has been devoted to WS standardization, mainly by four organization: W3C, OASIS, WS-I and Liberty Alliance. Following the recommendations of these standards, a WS can be an effective realization of a service according to the SOA paradigm: in a complex system every function is defined as an independent WS with well-defined invokable interfaces. Consider for example a web site of a book store, which permits buying and tracking the items sent to the buyer. The web application that manages all the operations related with the commerce can be expanded on several tasks, or on several services so that the whole process can be seen as a sequence of elementary tasks. Each service performs a well-defined function: the payment, the tracking service, the memorization of historical actions of the buyer, etc. Some functions can be realized by a single entity, but some others require external services such as the tracking service provided by the pony express service. In this framework, services can be realized with WSs, and they can communicate with each other by means of related communication technologies that permit the machine-to-machine interaction over a network.

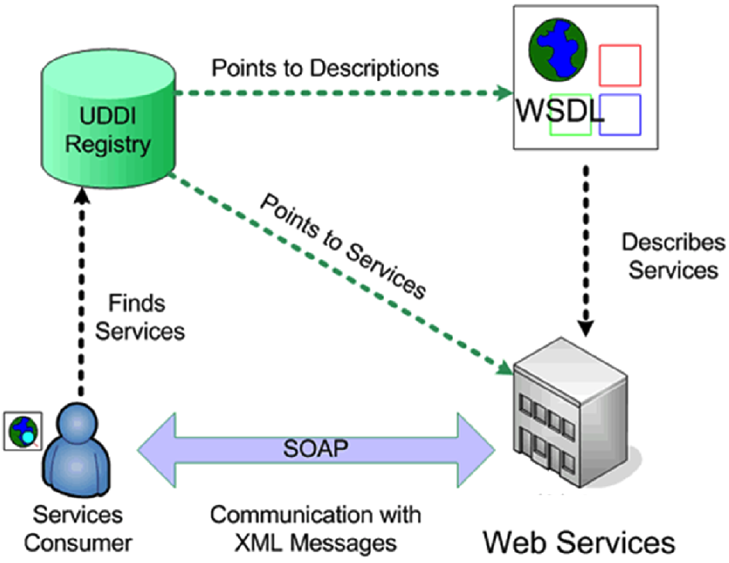

Figure 2.2. An example of Web Services functioning and correspondent SOA technologies used

In Figure 2.2 is represented an example of the Web Services functioning and the correspondent SOA technologies used. Web Services represent the prevalent way for the realization of services according to the SOA principles and are at the basis of the web applications we use every day, allowing for dynamically linking and reusing services provided by different parties over the Internet. Given the massive adoption of these technologies for the realization of the Internet applications, these introduce the following advantages in the NGN domain: there is a huge base of programmers using these mainstream programming technologies resulting in the development of many innovative services in the NGN; the operators may dispose of this base of skilled programmers for the development of the network service components as well as network control and management functionalities. By using the SOAP and XML standards for implementing service-oriented architectures the NGN will also implicitly benefit from the advantages of the SOA features previously described.

The evolution of the technologies related to the SOA in telecommunications over the last two decades is illustrated in Figure 2.3. Specifically, the main achievements aimed at supporting the deployment of web applications are found on top of the time axis, whereas the technologies specifically developed for the telco sector are shown on the bottom.

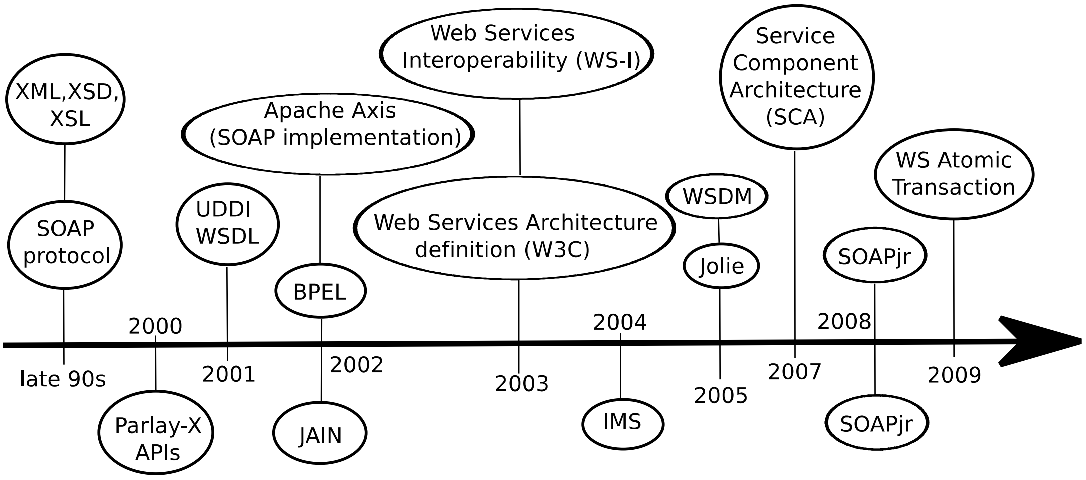
XML,XSD, XSL
SOAP protocol
Apache Axis (SOAP implementation)
UDDI WSDL
BPEL
Web Services Interoperability (WS-I)
Web Services Architecture definition (W3C)
WSDM
Jolie
Service Component Architecture (SCA)
SOAPjr
WS Atomic Transaction
late 90s
2000
2001
2002
2003
2004
2005
2007
2008
2009
Parlay-X APIs
JAIN
IMS
SOAPjr

Figure 2.3. Timeline of SOA-related key technologies for the Web Services deployment (on the top) and for the telecommunication sector (on the bottom).

In the late ’90s, the first SOA technologies were the eXtended Markup Language (XML) and the Service Oriented Architecture Protocol (SOAP). XML is a set of rules for encoding documents in machine-readable form. SOAP is a protocol specification for exchanging structured information in WS, which are the main technologies for the deployment of the SOA principles.

In 2000, the Parlay-X Web Service Application Programming Interfaces (API) have been defined, which apply to the fixed and mobile telephone network and exploit the advantages of the Web Services (WS) technologies to deploy an open telecommunication services market. In this way, the Parlay-X set enables software developers to exploit the capabilities of the underlying network. In the following years, other technologies related to the WS have been developed: the Universal Description Discovery and Integration (UDDI) and the Web Service Description Language (WSDL) in 2001; the Business Process and Execution Language (BPEL) and the first open source SOAP implementation in 2002. All these technologies contributed to the implementation of important features of the SOA paradigm, such as: abstraction and discoverability of the services and their simple composition and orchestration.

In 2002, the Java APIs for Integrated Networks (JAIN) has also been realized, aimed to make telecom service implementation easier than with traditional Intelligent Network (IN). In 2003, the World Wide Web Consortium (W3C) defined the functional components and the architecture of WS and in the same year the open industry Web Services Interoperability organization (WS-I) established the best practices for the interoperability between different WS standards. In 2004, the IP Multimedia Subsystem (IMS) was released. It is a standard for a unified-service-control platform for converging fixed, mobile, and cable IP networks and represents a key service platform for emerging NGNs.

In 2005, the WS Distributed Management standard (WSDM), a Web services architecture to manage distributed resources, and the Java Orchestration Language Interpreter Engine (JOLIE) were released. JOLIE is a programming language based upon the service-oriented paradigm: in Jolie everything is a service and there is no distinction between a local and a remote service. A service can be moved or replaced in a different location by simply means of updating the new service location: this permits to have a dynamic and fully reconfigurable system at runtime.

In the years from 2007 to 2009, three technologies have been developed. The first one is the Service Component Architecture (SCA), a flexible model used for the creation of business solutions using service components, defined within the OASIS standardization body.

In 2008, the new SOAPjr protocol proposal has been released. The traditional SOAP protocol, based on XML, appears expensive to parse and manipulate messages because of its verbosity. So, it consumes bandwidth and it is slow, especially on mobile or embedded clients. In contrast to the traditional protocol, SOAPjr (JSON-RPC) adopts a simplified coding that makes messages lightweight and easy to manipulate. Currently, an Internet-Draft RFC is under development.

Finally, in 2009 the WS-AtomicTransaction (WS-AT) protocol by WS-I completed the set of SOA key technologies, which are now important components in the deployment of WS. This protocol takes care of manage atomic transactions between distributed applications such as transaction managers or resource managers. An example of atomic transactions could be the sequence of transactions involved during the use of a credit card: ringing up the sale, swiping the card through a reader, verifying the validity of the account associated with the card, ensuring that the account has sufficient available credit and producing a receipt. Each of these components is an atomic transaction, and if only a single atomic transaction fails, the whole sequence fails. The WS-AT defines how transactions are carried out, and so manages the atomic transactions involved in the entire process. The achievements shown in Figure 2.3 represent the most important technologies and standards that foster the adoption of the SOA model when developing web and telcos services.

2.3 SOA in Next Generation Networks

In this section we survey the use of SOA technologies in NGN network technologies. Our investigation follows the NGN functional architecture as defined by ITU-T, focusing on the service and transport strata of this model and, within each of these, it highlights the main logic functions that have been the subject of investigation according to a service-oriented approach. The important issue of network management is also considered.

In the ITU-T Functional model, when describing the architecture there are explicit references to SOA. Specifically, the ITU-T rec. M.3060 ``Principles for the Management of Next Generation Networks'' [13] states that SOA should be the architectural framework behind the NGN management architecture. In the ETSI functional architecture, even if there is not a specific indication to the use of SOA, it is easy to see that the construction principles of this architecture are inspired by the SOA principles.

From the literature analysis most of the research works related to the SOA technologies in NGN refer to the functionalities of the service stratum. This is due to the fact that investments in research and development at this level guarantee telcos a higher return on investments with respect to other network areas. Indeed, creating new market-driven applications reusing an extensible set of existing service components (which are often provided by third parties) is a highly desired feature at the service stratum. Differently, due to the intrinsic nature of the functionality related to the transport of bits of data from one location to another, the SOA feature of dynamically connecting distributed services can't provide significant benefits in this framework. At this stratum, most of the attention has instead been devoted to the control and management of the transport services rather than providing the transport service itself.

From this analysis it can be seen that the service oriented paradigm does not introduce any additional service in the NGN networks that cannot be provided with the more traditional static layer-based approach. For instance, the functionalities provided by a SDP (Service Delivery Platform) developed according to the SOA principles are the same of those provided by the first generation (no-SOA-based) SDP. The benefits of using the SOA approach are in the interoperability, discovery, deployment, and re-use of the services/functionalities, which are tasks that are more easily implemented and managed when using SOAs.

This section is organized in three subsections that reflect the ITU-T functional architecture: the service stratum, the transport stratum and the management technologies.

2.3.1 SOA in the Service Stratum

Most of the research works regarding the SOA technologies in NGN refer to the functionalities of the service stratum. This is due to the fact that investments in research and development at this level guarantee telcos a higher Return On Investment (ROI) with respect to other network areas. By reviewing the literature on the subject, it can be found six areas that have been affected by these technologies; the relevant works are discussed in separate sections in the following. The main objective of each of these areas are:

- Service Delivery Platform
- Application Programming Interfaces
- Service Reuse and Selection

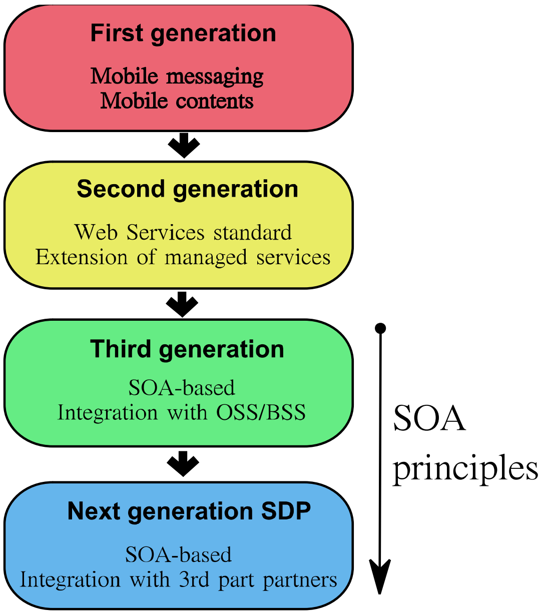

Figure 2.4. Evolution of Service Delivery Platforms

- Service Exposure environment
- IPTV in NGN
- IP Multimedia Subsystem (IMS)

Most of these areas are related to the management of the services and are strictly related each other. In particular, SDPs (Service Delivery Platform) are platforms that internally should incorporate a broker for service exposure and techniques for selection and reuse of services. IMS is a crucial architecture to provide multimedia applications in NGN whereas IPTV is expected to be the driver application provided by NGNs. These two architectures have to communicate with the SDP when creating new services and making use of specific APIs.

- ***Service Delivery Platform***

A Service Delivery Platform (SDP) is a platform that makes efficient creation, deployment, execution, orchestration and management of services possible. It can integrate legacy systems in an easy and flexible way and it enables third-party application developers to easily create and deploy various services for NGN. In [14], the evolution of SDP has been analyzed, and in the last 10 years four generations of SDPs can be identified, as shown in Figure 2.4.

The first generation of SDP was born as common service architecture for mobile services aimed at delivering mobile messaging and contents. The second generation extended these functionalities of service delivery to incorporate voice, multimedia, location, presence and charging services. In addition, it introduced the 3rd part access by using the new WS standards. The third generation of SDP is based on a layered architecture, where the SOA-based service orchestration and management functionalities have been introduced. This allows for an easy management and composition of distributed services to delivery complex applications over the network. This generation also integrated the delivery platform with the OSS/BSS (Operation Support System/Business Support System) systems, which are devoted to monitoring, controlling, analyzing and managing the network and to run its business operations towards their customers. The fourth generation, still following the SOA principles, focuses on cost effectiveness and reduced time to market for delivering highly customizable bundles of convergent services. System architecture is tuned for delivering services, which are aggregations of convergent services rather than monolithic services. This last generation, also called the Next Generation SDP (NGSDP), makes the possibility of 3rd-part partners to develop end-user services once and deploy many times easier. Additionally, a user can also become a content or service provider using the Web2.0 environment, which represents a set of technologies that speedup the development of web applications and that are extensively used in the current Internet.

Summarizing, the NGSDP is expected to be able to:

- integrate and interoperate with various networks such as mobile networks, wireless network, all-IP networks and so on;
- support standard technologies (protocols, middleware, API, etc);
- offer low cost, low risk and faster Return On Investment;
- reuse some service components to create and assemble new services (SOA-based);
- provide an Open API platform and support for easy creation of applications.

However, there is no a standard definition of SDP and NGSDP in the industry and in standardization bodies, although the TeleManagement Forum (TM Forum) [15] is currently working on defining specifications in this area. Additionally, limited research has being carried out on this topic. One of the most interesting research works is that presented in [16], which defines an SDP architecture that tries to incorporate the main features listed in section 2.2 for SOA-based systems, with particular attention to: permission to modify or create new services by adding or deleting single service components; simple customization to easily add new functions as service components to commercial products; support for multivendor environments.

Many vendors have developed proprietary SDP solutions, such as: Ericsson Drutt MSDP, Zeus ZXTM, Nokia Siemens SDF, Oracle SDP family, HP SDP, just to cite some of the major ones. Indeed, all the major vendors and consulting enterprises provide their own solutions given the importance of this component for a prompt reaction to the market needs. Some solutions are designed for a specific environment, such as the Ericsson MSDP devoted to the mobile world, whereas others apply to any type of service. They all believe that SDP design and build processes using SOA compliant subsystems have the advantage of evolving gradually, with the effect to maximize the benefits and minimize the risks associated with each step.

- ***Application Programming Interfaces***

An Application Programming Interface (API) is an abstraction that defines and describes an interface with a set of functions used by each component in a software system. In the telecommunication field, the APIs permit the development of new telco services without possessing special competencies in the underlying technologies.

APIs designed to allow 3rd party developers access to the NGN platform are also called Open APIs. Hence, developers, and so applications, can have access to core network functionalities using open standardized interfaces. There is a great interest in Open APIs from the operators since these stimulate the development and deployment of applications over their networks and then increase the network service utilization. Security problems may obstacle this process and for this reason, since the beginning, all the defined APIs have included authentication and secure communication technologies, as illustrated in [17].

One of the first standardization efforts of telco Open API has been the release of the Parlay group APIs [18], a multivendor consortium founded in 1998 that works closely with ETSI and 3GPP. This first series of APIs was strictly limited to the telephone network and included the functionalities of call control, conferencing, user interaction and billing by using web services. In 2003 and subsequently in 2006 (Version 2.1), the Parlay group released a new set of web services SOAP-based API, that is the Parlay-X. The pending version 3.0 introduces message broadcast, geocoding, application-driven quality of service, device configuration and multimedia streaming/multicast. A joint effort between ETSI, 3GPP and the Parlay group has brought to the Open Service Access (OSA)/Parlay Framework specification [19]: it provides secure, controlled access to network capability provided by a network operator to 3rd party service providers. OSA/Parlay APIs expose almost all the network capabilities provided by the corresponding network protocols and makes combination of several service capabilities and integration of IT applications easier. Some other standardization bodies are working on specifications of APIs and web services. Some define generic APIs, such as the OASIS, W3C, and Liberty Alliance, some others more specific to the telco sector, such as OpenIPTV forum and OMA (Open Mobile Alliance). However, Parlay-X is the best-known web-telecom API for publicizing telco functions. Definitely, since the first release the Parlay standards have followed the SOA principles and technologies. The main reason has to be found on the widespread acceptance of the WS technologies and on the SOA benefits of allowing for an easy integration and reuse of services developed and provided by different operators, which is one of the main objectives of the Open APIs.

Other than standard APIs, industries have defined their own proprietary Open APIs and together with the interfaces definition they usually provide the Software Development Kits (SDKs) [20]. The first company that published API/SDK is British Telecom in 2007. In the following years, other European companies such as Orange, Deutsche Telekom, Telefonica and Ericsson published their API/SDK to provide telecommunication services. All these products are SOAP-based APIs and most of the time similar to Parlay X specifications; in 2008, British Telecom also published a SDK based on Flex (Adobe Flash with ActionScript) library for graphical Internet applications. NTT Corporation also developed a web-telecom SDK to facilitate use of Parlay X. From the existing products it arises that neither proprietary nor standard APIs have prevailed over the other, since each solution has its own well-known pros and cons. Standard ones are advantageous for the application developer since his applications can work in several environments without significant changes. The proprietary solutions represent a good approach for the operator/vendor with a high market share since the applications working with his environment are not able to run in others.

Research on Open APIs mostly focused on the main recommendations that should be followed during the definition process in the standardization bodies. Important recommendations are outlined in [21], which states that while significant progress has been made in the standardization of NGN architecture and protocols, little progress has been made on successful open APIs. One of the most important considerations is the need to release royalty-free APIs to attract the biggest number of developers to the NGN world. Unfortunately, most of the specifications and the APIs are released with the payment of a royalty. For example, in 3GPP and Open Mobile Alliance (OMA), the use of the released API is ruled by the specific intellectual property rights (IPR) adopted by the organizations belonging to the OMA alliance. Indeed, Parlay-X API is the only major standard that has been released royalty-free: as a result, these APIs are the most accessible API that are available to 3rd party developers. Another important consideration raised by [21] is that, in the process of standardization, vendors and operators are in charge of retrieving feedbacks from developers; this implies that all feedbacks from the developer community are filtered by the telecom operator or vendors and adapted to their vision of where the industry should go; essentially, the standards are driven heavily from a commercial viewpoint and feedback from developers is filtered from this perspective. The effect on the entire process is that the speed with which specifications are currently created is greatly reduced, being one of the main limits. In [22], authors suggest the adoption of open source methodologies within the standardization process, with the aim to increase the confidence of the network operators on 3rd parties developers.

- ### *Service Reuse and Selection*

Nowadays, services and applications are composed of a large number of functions combined together, that involve different resources of a telecommunication network. Due to this complexity, the network is subject to failures caused by many factors such as communication, hardware or software problems. Failures may impair the availability of the whole business process execution. The simplest reaction in this case is the discovery of the failed service and the substitution with an alternative one at run-time.

The management of each service component to prevent and handle failures is crucial for the overall system performance. The exact definition of the activities features that have to be carried out in this context are still to be clearly defined so that there is not a universally-accepted definition. For that reason the market is missing specific products and standards on this subject. Differently, a quite significant research activity has been carried out in recent years, dealing with: how to evaluate the degree of reusability, how to discover the appropriate service, how the selection of a service affects the overall Quality of Service. In all these activities, the SOA principles are getting more and more attention.

The reusability of services is defined as the degree to which the service can be used in more than one service application or business service, without having much overhead to discover, configure and invoke it. In [23], the authors recognize the importance of the SOA model for publishing common features as services and reusing the published services in building applications. They then focus on a model evaluating reusability of services, applied to both atomic and composite services. Five quality attributes of reusability have been defined:

- Business Commonality: it measures the degree to which functionality and non-functionality of the service are commonly used by the consumer in a domain;
- Modularity: it measures the independence functionalities of a service, without relying on other services;
- Adaptability: it is the capability to be well-adapted to different service consumers;
- Standard Conformance: it is the degree of compliance with widely accepted industry standards;
- Discoverability: it measures the fidelity of service descriptions and specifications to concrete and tangible service features.

For each of these attributes a formula has been defined to measure their extent. The resulting expressions have then been combined in a reusability quality model. In [24], the author describes some best practices to be adopted for improving the service reusability:

- decouple the physical transport from the service logic, in each service;
- provide standard interfaces for service access;

- create service adapters for backwards compatibility;
- ensure that services are interoperable.

As to the service discovery issue, a few important questions should be considered:

- how to describe user requirement and the abilities of existing services;
- how to efficiently discover the services that meet user requirement;
- how to response the case that there is no service matched to user requirement.

In [25], the service discovery is considered as an essential support function for a service architecture operating over multi-domain and multi-technology networks. In this work, the authors have surveyed several service discovery strategies with the aim to identify the characteristics suitable for large-scale and multi-domain networks. They have focused on scalability, standardization, performance, effectiveness, system independence and implementation support attributes. The Web Services approach appears as one of the most relevant and potential component of the proposed architectures.

Leveraging on SOA principles, in [26] the authors investigate the automation of the service selection, developing a model based on the expectancy-disconfirmation theory: this one relates expectation, perceived quality and quality disconfirmation. To this, some user satisfaction forecasting models are derived: the customer expectation is assimilated to the contracted quality level (agreed QoS) and the perceived quality to the measured one. A negative disconfirmation between the agreed QoS and the measured one leads to the dissatisfaction of the user and so to a negative feedback: the feedback reflects the satisfaction of the user with the service. The feedback computed is used to evaluate the trustworthiness and credibility of services offered. The automation of the rating process frees the users from the rating task and ensures the objectivity of feedback. In the same service selection topic, [27] proposes a SOA-based service description and selection framework to improve the accuracy and the performance of service selection in the industry. The most important task in this framework is the service classification process, which is then used in a service ability matching algorithm, computing the satisfaction level of available service implementations on the basis of the service requirements. Also [28] proposes a strategy to select and recommend services based on multi-QoS constraints. The system focuses on measurable QoS attributes, i.e., performance, reliability, and availability. In addition, these attributes are dynamically updated with an Artificial Neural Network (ANN) to satisfy the customer preferences.

- ***Service Exposure environment***

A service exposure environment is based on an intermediation system that decouples the offered services from the providing infrastructure. The main benefits of adopting this solution is to allow for an easy discovery of services and to facilitate the interoperability between the service requester and service implementation, whose technical working details are hidden from each other. In this application scenario, an example of a SOA-based product could be an entity that works such as a service broker. The service broker responsibilities can be simplified in the information collection about existent and available services in different administrative domains. In addition, the service broker negotiates the use of each service with the service owner (i.e. how many times a service can be used in a day, how many services a client can use from the same service provider or the price related to each single use).

In a service exposure environment, the adoption of the SOA paradigm and of the service broker can facilitate the interactions between services; at the same time, the application developers can obtain detailed information about all SOA-based services that are easily available in the current administrative domain or in other domains. There are no specific standardization activities aimed at defining the protocol and interfaces of service exposure systems, whereas research is quite active in this field, addressing the expected functionalities and architectures. Additionally, as is described at the end of this section, mature products are available in the market.

In [29], the authors propose an approach for the development of a service broker in a policy-based service exposure environment. They believe that even if the WS technologies provide a flexible and loosely coupled means of integration, the actual interfaces descriptions are not understandable by businessmen. In fact, there is a gap between the high-level business services and low-level services implemented with WSs. The proposal, named Intent-based Service Request API (InSeRt), which follows the suggestion in [30], is related to the changeover from the function-driven SOA to intention-driven SOA. In this way, the service interfaces highlight the business goal that a service is able to achieve not the method signatures or classes that are invocable. The intent-based service request setup follows three steps: the first is the intent expression or description format, the second is the service capability description, and the last is the matching and composition evaluation that coordinates with the intent and service enablers. The major advantages of this approach can be summarized in two facts: easier access to services by business people and the ability to dynamically combine enablers on the basis of the business goal.

Another approach finalized to overcome barriers between application level and Information Technologies (IT) experts is the eXtreme Model Driven Design (XMDD) [31], a technique that puts the user-level process in the center of development to reach customer satisfaction. In fact, a customer or user can create and control the whole life-cycle of their activity. The jABC approach [32] is an implementation of the XMDD approach, which offers an environment that allows the customer to define the service features with any desired detail level. Against the logic of traditional tight telecommunications service definition that has problems of organizational inertia, static routines and inefficient path dependencies, the jABC approach supports pluralistic interfaces: each functional activity defined by the user, here called Service-Independent Building Block (SIB), is associated to a software implementation via APIs. A single application is finally composed by service orchestrations realized in terms of SIBs and designed as Service Logic Graphs (SLG). In [33], a realization of a SIB library for Parlay-X (Open APIs for telco services presented in Section III.B) services has been developed. Each Parlay-X interface is composed by one or more SIB that provide the underlying enabler functionality in an abstract manner for service modeling inside jABC: all the functionalities are available and can communicate with a Parlay-X gateway thanks to Web Services. The tools used are the open source Apache Axis2 and Java Web Services engine. This framework permits exposing NGN to the web in a seamless way: the Parlay-X services become available for a high level mash-up design, which is the process frequently used on the Internet to provide complex web application by linking distributed (mash-up) web services.

In the industry, products [34] try to eliminate the need for custom, proprietary, and manual process of exposing network APIs and on-boarding third-party partners and applications. In this way, a service exposure platform provides a comprehensive portfolio of out-of-the-box, converged Web-SOA Telecom interfaces, and application programming interfaces to accelerate third party application development and integration.

- ***IPTV in NGN***

In the context of NGN, triple play and quadruple play services employ one single medium via IP for the provisioning of all these services. The triple play is defined as the service where broadband data connection, voice, and television are provided by a single operator over a single cable. In quadruple play the mobility feature is introduced so that the services are also provided over wireless channels. In this scenario, the IPTV is expected to be the main driver, whereas new value-added services will be introduced also using the IP return channel from the user side. Basically, the new services should have the features of personalization, interactivity and community relations and following these guidelines many new concepts can be built. In this framework, a SOA architecture is desirable, because it can assure an effective composition and orchestration of all services required. In [35], authors recognize the adoption of elementary building blocks (i.e. Web Services) for the development of complex media workflows as an important step towards the reuse of media processing tools in an IPTV domain.

The introduction of a SOA-based framework for the integration of multiple sources of data and content would enable IPTV providers to use a loosely coupled and federated architecture, with the advantage to easily integrate reusable and vendor independent services. Inspired by these principles, in [36], the authors have shown how IPTV services can be decomposed into common functional blocks. Those blocks get wrapped up with a webservice interface, allowing them to be composed and orchestrated with other webservices to form new services within platforms for serviced delivery and on top of complex NGN telecommunication infrastructures. This constitutes the basis for a quick and dynamic provisioning of customizable telecommunication and integrated multimedia services, which are required by the customers that become more and more persuaded by new ways of interaction with the provided content. The technique to decompose a known service into its basic function, then giving those a webservice envelope, can be seen as a general technique to enrich the portfolio of existing webservices. This, of course, results in a nearly limitless variety of services to be constructed, especially when it comes to integrate well-known services (such as: TV + Advertisements + Telephony + Shopping + content sharing), which gives a customer the same look and feel like from the single services.

Unfortunately, [36] has not provided experimental evaluations of these benefits, which are also quite intricate to be assessed. However, the proposed functional block decomposition has been conducted bearing in mind well-defined real use-cases, such as: multimedia shared experience, where the users can share their owned content with other users; community TV, that provides the way to create communities of fans of specific video contents; content tagging, by means of which the user can add a tag (or a comment) to a dedicated scene within the content stream and publish this tag to all users within the user's community; customized advertisement; interactive TV, that allows the user to interact with some video programs, such as the participation to quiz-shows. [12] also recognizes the benefits of adopting the SOA paradigm when defining the evolution of the IPTV architectures. This paper analyzes the migration of the current Non-NGN IPTV systems towards the NGN converged IPTV following the TISPAN NGN release 2, which allows for the interoperability with IMS and non-IMS architectures. All the possible evolutions analyzed follow a service-oriented paradigm and each solution is compared in terms of the number of functionalities provided. Part of the proposed architectures has been implemented and experimented showing the effectiveness of the designed solutions. However, also in this case, there is no quantitative evaluation of the introduced benefits with respect to non-SOA architectures.

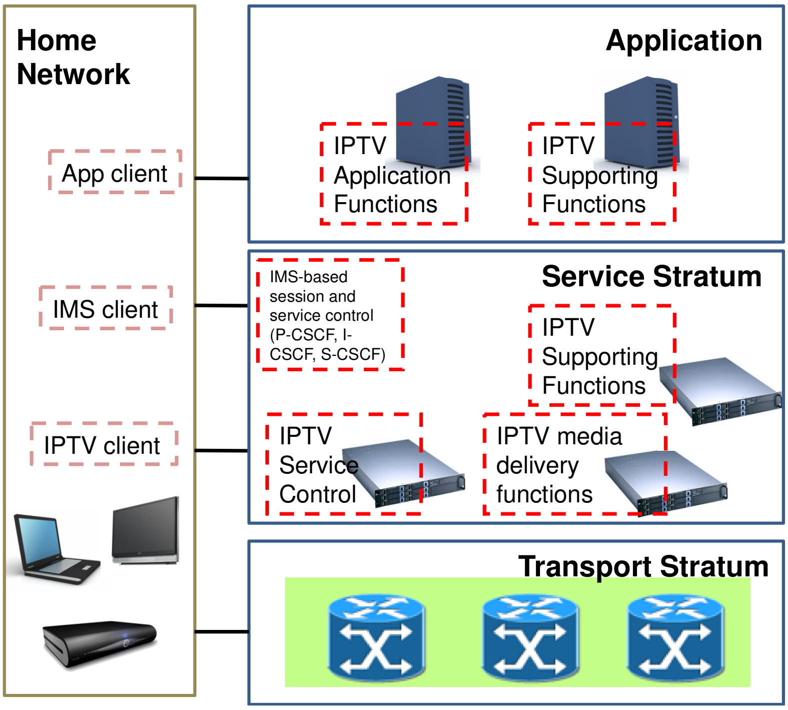
Home Network
App client
IMS client
IPTV client
Application
IPTV Application Functions
IPTV Supporting Functions
Service Stratum
IMS-based session and service control (P-CSCF, I-CSCF, S-CSCF)
IPTV Supporting Functions
IPTV Service Control
IPTV media delivery functions
Transport Stratum

Figure 2.5. IMS system architecture and its relations with other NGN components

- ***IP Multimedia Subsystem (IMS)***

The IP Multimedia Subsystem (IMS) is an architectural framework for delivering Internet Protocol (IP) multimedia services. Figure 2.5 shows a simplified architectural model of IMS and its relations with the other components of an NGN network. In this scheme the IPTV servers have also been included. From the figure it is clear that IMS is a collection of different functions, linked by standardized interfaces. A function can be implemented by one or more physical servers so as to split a single function into two or more nodes. Each node can also be present multiple times in a single network, for dimensioning, load balancing or organizational issues. The Home Subscriber Server (HSS) is a user database that supports the IMS network entities that actually handle the calls. It contains the subscription-related information, performs authentication and authorization of the user, and handles the user location information. Several roles of Session Initiation Protocol (SIP) servers or proxies, collectively called Call Session Control Function (CSCF), are used to process SIP signalling packets in the IMS. The SIP is an important standard used for the setup and control of application session for the TCP/IP application layer. These servers (the P-CSCF, I-CSCF and S-CSCF) are responsible for handling the session signaling from the user and with other IMS domains and communicate with other servers that provide specific multimedia services. In this figure we show the connection with the IPTV media control and delivery functions.

The research and the standardization bodies in the field of IMS are active and newer specifications are periodically released. The IMS has experienced a long standardization schedule since its birth. In 1999, the 3rd Generation Partnership Project (3GPP) [37] released the first specification. In the following years, IMS was also introduced in UMTS (2003) and a standardization body of ETSI, Telecommunications and

Internet converged Services and Protocols for Advanced Networking (TISPAN) standardized IMS as a subsystem of NGNs (2005). The last release was in 2008, when some elements to the NGN such as IMS and non-IMS based IPTV, Home Networks and devices, as well as NGN interconnections with corporate networks were added. [38] discusses the characteristics of SOA and why SOA principles can improve IMS. The loose-coupling, the interoperability, the modularity and the well-defined SOA interfaces which are independent of the underlying implementation, make the exposition, redesign and integration of IMS functionalities possible. In the IMS architecture, native or legacy telecom services such as SMS/MMS messaging, and Next-generation services such as SIP-based call control, can benefit from their integration in a SOA architecture, also opening IMS architecture to the 3rd-party application provisioning.

The possibility of using SOA to integrate IMS-based enterprise communication systems within enterprise information technology (IT) infrastructure is analyzed and tested in [39]. Considering a simple IMS scenario, the central node which takes care of the integration task is the IMS application server. So, all coordination efforts between the application server and the other IT systems are realized by means of Web Services, and the communication happens with SOAP messages sent over the service bus.

In [40], the authors study the roles of an SDP and IMS. Even if at the beginning the IMS was retained as a competitor of the SDP, now they are considered as complementary and the industry sees them as key technologies in the evolution towards NGN. Moreover, the authors predict a new era for service delivery platforms based on SOA and integrated with IMS; the adoption of the service modularization and the reusability introduced according to SOA guidelines are expected to provide a highly efficient interconnection between service delivery platform and IMS. In the industry, many commercial products that embrace IMS and SOA are available. However, even if vendors recognize the power of SOA applied to IMS, they introduce the SOA by means of additional software, following an evolutionary approach which guarantees the support of legacy IMS systems. The IMS conversational transactions can be easily included into SOA services, since every transaction can be seen as a SOA simple service that can be composed to provide a complex IMS service realizing a task in a business system. Due to their characteristics, IMS transactions are well and quickly deployed in a SOA framework and vendors such as Service Oriented Legacy Architecture (SOLA) [41] and [42] develop products able to efficiently handle IMS transactions.

- ***Considerations about SOA in the Service Stratum***

From the analysis presented in this section we may draw the following considerations:

- SOA principles represent key guideline in the development of SDP solutions, allowing for a quick and dynamic provisioning of the services to the users. Indeed, the latest platform (NGSDP) integrally incorporates the major SOA technologies. The research activity in this area is limited, whereas all the major vendors provide advanced NGSDP solutions;

- The OSA/Parlay is the Open API that most probably has (or will have) the major impact in the deployment of NGN. Since the first released standard, the SOA principles have been followed for the definition of the interfaces between services provided by different operators and application developers/providers. SOA-related technologies are used to develop relevant Software Development Kits, which allows for quick developments of applications making use of the reference APIs. Research on Open APIs mostly focused on the main recommendations that should be followed during the definition process in the standardization bodies. One of the most important considerations is the need to release royalty-free APIs to attract the biggest number of developers to the NGN world. Indeed, most of the current specifications now are being released with the payment of a royalty;
- In recent years an increasing interest in functionalities dealing with service reuse and selection has been observed. However, this is a subject that still needs to be investigated in-depth and for this reason there is not a universally-accepted definition and the market lacks specific products and standards. Differently, a quite significant research activity has been carried out, which proposes the use of SOA principles to: evaluate the degree of reusability; help the discovery of appropriate services; and drive the selection of a service on the basis of the impact on the overall Quality of Service;
- Service exposure environment is devoted to making services visible and easily accessible internally and to third parties. Here WS technologies provide a flexible and loosely coupled means of integration, but the actual interfaces description are currently not understandable by business people. This is the major problem addressed in this area.
- IPTV is expected to be the main driver for NGN deployment, fostering the provisioning of new value-added services. The new services should have the features of personalization, interactivity and community relations. In this framework, a SOA architecture is desirable, because it can assure an effective composition and orchestration of all services required.
- The IP Multimedia Subsystem is a crucial component in the NGN service stratum, with newer specifications periodically released by the 3GPP. The importance of the SOA

technologies in the development of the IMS systems is proved by the extensive set of available SOA-based products.

2.3.2 SOA in the Transport Stratum

The transport stratum is aimed at providing two main functionalities: the transport of data from one end of the network to the other, traversing different access and core networks, according to the level of QoS requested by the end-user; and the control and management of the transport functionalities, which should allow for an easy and robust activation, monitoring, updating, and configuration of the transport services. It is a matter of fact that the IP technology is and will continue to be the core of this Stratum, with all the enhancements required to improve the QoS capabilities, the security of the communications and the simplicity in service management, such as the Differentiated Services (DiffServ), Multi Protocol Label Switching (MPLS), IPv6 standards and many others.

Due to the intrinsic nature of the functionality related to the transport of bits of data from one location to another, the SOA feature of dynamically connecting distributed services can't provide significant benefits in this framework. Indeed, in this case, we are talking about interconnecting intermediate systems to create the physical path connecting the communication end-points, with associated geographical constraints that limit the choice of the apparatus to be used to deploy such a transport service. Differently, when we deal with the control and management of the transport services, again, we can benefit from the reusability, coupling abstraction and composability features of the SOA. Think of interconnecting services for the monitoring of the transport service, the management of the QoS policies, the computation of a route and many other functions, that have only limited location constraints and can then be implemented according to a service-oriented approach.

In the following section, the SOA works are grouped in three areas:

- Transport service management
- Access to transport devices and services
- Considerations about SOA in the Transport Stratum

- ***Transport service management***

A few papers have addressed the design of transport service management solutions making use of the SOA paradigm and technologies. We believe that the proposed works still need to be consolidated, especially with on-field trials and validation. It has to be noted that the proposed solutions focus on the MPLS protocol, even if the basic principles could be applied to other ones. This is probably motivated by the fact that MPLS is recognized as one of the most powerful protocols that allows for adding traffic

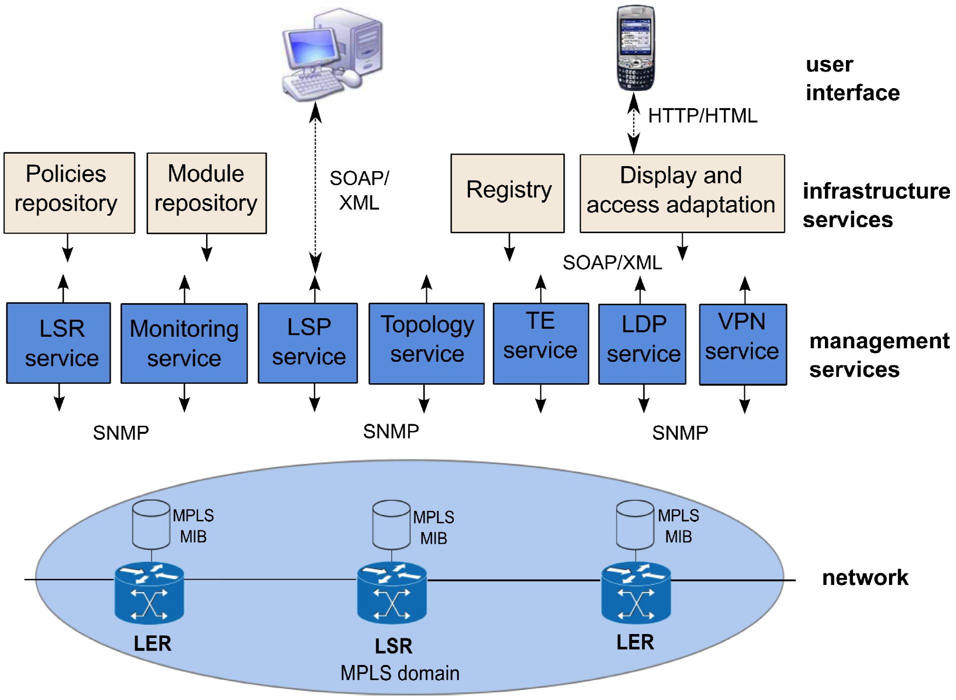

Figure 2.6. Distributed and web service-based universal architecture for MPLS networks

engineering functionalities to the IP world and to manage QoS when combined with other technologies. Note that traffic engineering and QoS are two key functions that are expected to be provided by NGNs. Due to the fact that limited effort has been devoted to this issue until now, only research and experimental works are available whereas products and standards are missing.

In [43], a distributed web-based approach for the management of MPLS networks is proposed. In an MPLS network, data packets are assigned labels and packet-forwarding decisions are made solely on the contents of this label, without the need to examine the packet itself as is done in traditional IP networks, allowing for the creation of end-to-end circuits across any type of transport medium. To guarantee the adequate level of QoS in the Internet network, the end-to-end paths are selected on the basis of some cost functions aimed to, for example, reduce costs, balance network load, or increase security. Specific routing algorithms, i.e., the Constraint-based routing algorithms, select a routing path according to either a service-oriented (QoS-based) or administrative-oriented (policy-based) constraints [44]. The management architecture proposed in [43] consists of four layers: the user interface, the infrastructure services, the management services and the physical network, as shown in Figure 2.6. According to this architecture, through the user interface the administrator accesses either the infrastructure or the management services: the first represents an abstraction of the complex and potentially network-wide services, such as the policy repository or the registry service, whereas the second accomplishes specific management tasks, such as, the topology discovery, the monitoring and the LSP (Label Switched Path) setup. The management layer communicates directly with the network layer, which represents the lowest layer made of physical devices. This architecture has been implemented focusing on the MPLS transport technology over IP networks [45]. In particular, the implemented management services are the following:

- Virtual Private Network (VPN) service
- Label Distribution Protocol (LDP) service
- Traffic Engineering (TE) service
- Topology service
- Label Switched Path (LSP) service
- Monitoring service
- Label Switched Router (LSR) service

Each one of these services can communicate with the upper layers using SOAP/XML messages and with the physical devices using Simple Network Management Service (SNMP): in each device a SNMP agent is supposed to be active. The architecture proposed combines the advantages of conventional distributed approaches with the SOA principles realized in a web-based framework. Management and infrastructure services can cooperate exchanging data according to the standard XML: in this way the interoperability of different vendors devices and of different management platforms is supported. Additionally, the use of SOA principles and formalism allows for the construction of an easily expandable system.

Whereas this solution combines the use of SNMP with SOAP, other studies have compared the performance of the management systems when using these two types of technologies. In [46] and [47], the WS composition from the network management point of view has been addressed following two procedures. The first one is the device network aggregation, which is desirable when many types of information are collected from a single device. The second one is the network information aggregation, which is desirable when the information is aggregated from different devices over the network. Both these approaches can retrieve information very easily because only a web service call is needed: following this technique, the amount of aggregated information increases, as well as the saved bandwidth, because the number of SOAP requests and replies is reduced. However, the most important advantage is related to the capability to retrieve information from devices in different administrative domains. Usually, in this case the SNMP experiences its limits because SNMP traffic is confined to a single administrative domain and so it is blocked by border firewalls. Associating a WS very close to the SNMP agent in each device, the last one becomes reachable by every device in the Internet because SOAP over HTTP/HTTPS messages are usually not blocked by firewalls.

- ***Access to transport devices and services***

An issue strictly related to network management is that of the access to the transport devices for monitoring and configuration purposes. In this section we analyze the main works that have been proposed with reference to this topic and that make use of a service oriented architecture approach. In these works the MPLS is not a predominant protocol since the focus now is not the management of the data traffic but how to make the communication with the physical infrastructure by the network administrators efficient and reliable. To this purpose, the NETCONF protocol has been defined by the IETF, which is first reviewed in this section followed by an analysis of research work on using WS for device access.

The NETwork CONFiguration (NETCONF) protocol [48] defined by the IETF NETCONF working group is specifically devoted to network management and permits the handling of individual network devices, as well as set of devices that are offering a certain IP service. In this contest, the IP service can be an IP connectivity service (i.e., the configuration of routing protocols or IP/MPLS VPNs) or a higher-level service, such as the VoIP service. The NETCONF is structured into several layers: each one is responsible for a functional part of protocol operations. A management application (NETCONF client), and a managed device (NETCONF server) participate in the exchange of information according to a request/response scheme using the Remote Procedure Call (RPC) paradigm.

The NETCONF protocol is a transport-independent protocol that can be activated using any transport protocol that satisfies the following requirements: being a connection-oriented protocol, because a persistent connection is required between NETCONF peers; support authentication and integrity mechanisms, using secure channels for sensitive data transport and validation of remote peers. Actually, the NETCONF implementations support the main protocols for the deployment of WS (SOAP,[49]) and for secure connections (Secure Shell, SSH, [50]). The messages exchanged are encoded according to the open XML standard: new operations and functionalities can be defined, specified and implemented according to the vendor or new technology devices or protocols. The NETCONF protocol specifies nine operations, that are invoked to retrieve, install or modify configuration data of a given managed device.

Summarizing, the NETCONF protocol is an easy management protocol that separates configuration and state data, and thanks to the adoption of XML encoding, SOAP/HTTP transport protocol and the request/response scheme such as RPC, permits operators to concentrate on the configuration of the whole network rather than on individual devices. Even if not totally compliant with the SOA principles, in a short time the NETCONF evolution will use the SOA concepts to extend the potentialities of the simple RPC communication protocol. Regarding the network configuration area, there are not standardization bodies active and on the market there are not specific commercial products.

As to the access of specific physical devices, [51] proposes a network management solution based on the SOA concepts. The desired final target is to reduce the gap between the services or applications that reside in the upper NGN layers and the real single devices, which inside a network contribute to delivery of the traffic data. In the past, starting from this need, many research directions have led towards the concept of network programmability, and in this sense the presence of programmable interfaces and more in general of a middleware able to communicate with software applications and directly with physical devices has emerged. The solution proposed in [51] permits full control of every network device using a Web server proxy, which is responsible for dispatching the control commands to the devices. Each WS exposes its own functionalities using an object-oriented library, hiding the details of the communications with the physical devices. The management information data are exchanged between various web services using the SOAP protocol, that if compared with classical SNMP protocol can have a significant impact on the network load due to the heavy overhead related to SOAP messages. For this reason, the authors suggest avoiding a one-to-one map of every functionality into a service method, because the best benefits can be reached with a sequence of multiple low-level commands to be dispatched to the physical device included in a unique method. The architecture proposed shows many advantages related to the fact that higher-level services can invoke and control every web service proxy without taking care of the inner details of a single device. The research is active in the area of the access to physical devices; standard specifications and commercial products are not available in this area.

- ***Considerations about SOA in the Transport Stratum***

From the surveyed works we draw the following conclusions:

- Service management is the main issue addressed at this stratum making use of the SOA principles. The presented solutions propose the adoption of the SOA technologies to implement an orchestration layer to support the deployment of complex network management services by combining simple and re-usable services. The main advantage is the interoperability among different vendors devices and platforms when managing transport services;
- SOA technologies are also proposed to the build an abstraction layer for accessing the networking devices, which relieves network operators from inefficient, device-specific micro-management. Indeed, WS are proposed for exposing the equipment own functionalities using a service-oriented library, hiding the details of the communications with the physical devices;
- The adoption of a SOA-based architecture does not require re-implementing all the transport services. Most of these would just need to be wrapped up by the required WS interfaces, some others may require being divided in simpler services and some

others grouped. The most important activities to be carried out is the definition of all the required simple services and their interfaces;

- The NGN transport technologies have to be analyzed and decomposed into elementary services. This process is not simple and costless. In addition, the transformation of a monolithic system in a modular system composed by several elementary services is a necessary task to obtain a SOA-oriented system design;
- Experiments still need to be carried out to quantify the benefits of using SOA-based architectures at the transport stratum.

2.3.3 SOA in Management Technologies

The SOA philosophy and technologies seem to be the ideal framework for the management and integration of heterogeneous technologies and systems, due to the support of loosely coupled services and the dynamic creation of ad-hoc applications built on the basis of existing services. The same ITU-T rec. M.3060 "Principles for the Management of Next Generation Networks" [13] states that SOA should be the architectural framework behind the NGN management architecture.

This section is organized in three parts:

- The evolution and the SOA-based implementation of Operational Support System (OSS)
- The Business Support System (BSS) solutions
- Conclusion

o *OSS and BSS architecture evolution*

OSS and BSS are two key components of NGN network management. The first is a set of programs that helps the communications service provider to monitor, control, analyze and manage a telephone or computer network, whereas the second is another set of components that telcos use to run its business operations towards customers. Even if these two platforms maintain their own data and service responsibilities, the boundary between them is not well-defined. As detailed in the previous Section II.B, the NGN functional model is characterized by the separation between the Service and the Transport stratum. In addition to this fact, the NGNs are intended to be made up of heterogeneous networks, introducing a high level of distribution and complexity in management. As a consequence, the introduction of NGN networks has made traditional OSS/BSS architectures inadequate, pushing for a quick evolution of the adopted solutions.

The advancements in OSS/BSS architectures are analyzed in [52], where three different stages of models are presented, as shown in Figure 2.7. At the first stage a simple manager-agent model is adopted: the manager uses the Management Information Base (MIB) in the Network Entities (NE), whereas the agent receives messages from the manager to read or change the network element status according to the SNMP protocol. At the second stage the management systems concentrate on the subsystems within their network: each technology adopted has developed an architecture and information models, that are managed by a specific Network Management (NM) entity (see Figure 2.7.b). Each subsystem needs a separate OSS and so an extra layer is required in order to obtain a network-wide management view, introducing more complexity in the management architecture.

Today, the architecture adopted by most telcos is the Enterprise Service Bus (ESB) (third stage shown in Figure 2.7.c). The main difference with respect to previous architectures is the presence of a middleware message bus between the OSS layers and between the OSS and BSS. All communications are provided through messages, and the architecture becomes distributed, loosely-coupled and supports integrated management capabilities. It is the first evolution towards a SOA-oriented design.

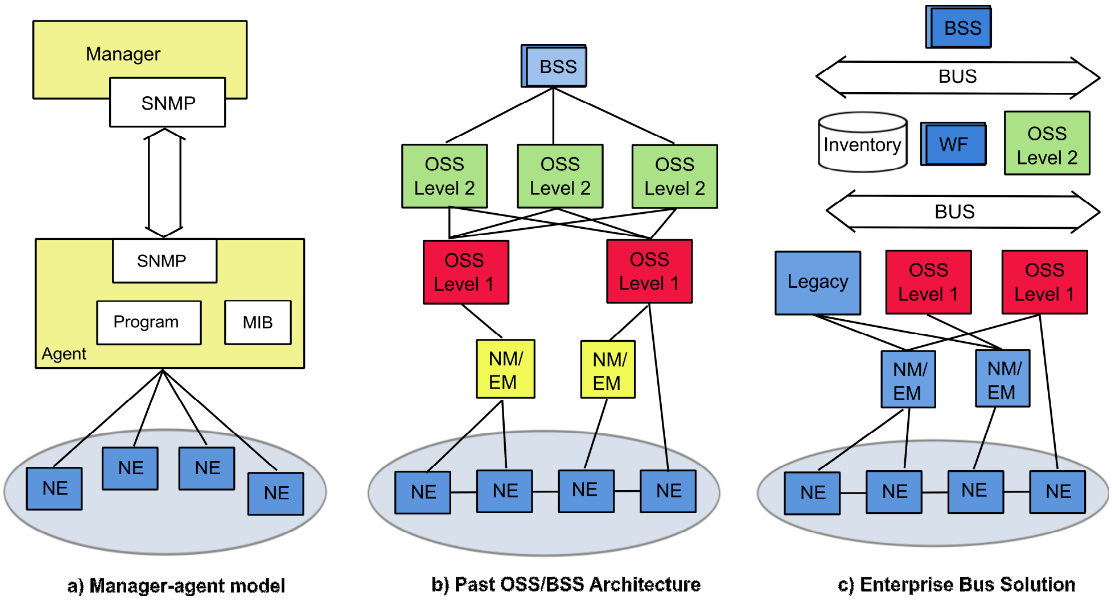

Figure 2.7. Stages of OSS/BSS evolution: a) the Manager-agent model, b) the Past OSS/BSS Architecture and c) the Enterprise Service Bus (ESB) solution.

- ***SOA-based OSS/BSS***

The New Generation Operations Systems and Software (NGOSS) [53] is an initiative of TM Forum (TMF), which is based on the architecture of the third stage described in the previous section. To improve

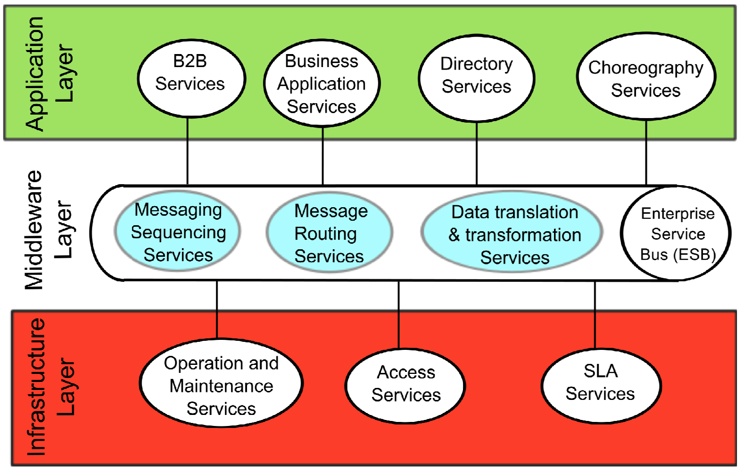
Application Layer
B2B Services
Business Application Services
Directory Services
Choreography Services
Middleware Layer
Messaging Sequencing Services
Message Routing Services
Data translation & transformation Services
Enterprise Service Bus (ESB)
Infrastructure Layer
Operation and Maintenance Services
Access Services
SLA Services

Figure 2.8. SOA-based NGOSS architecture

the convergence of telecom and enterprise management, a collaboration started between the TMF and the Distributed Management Task Force (DMTF); in particular, these organizations investigate the integration of Shared Information and Data Model (SID) and the Common Information Model (CIM). The former defines common vocabulary and a set of relationships used in the definition of NGOSS architectures, whereas the latter is a technology-independent information model used to describe enterprises management interfaces in DMTF. The resulting solutions make intensive use of the SOA philosophy, which then drives the further evolution of the architecture,into the scheme sketched in Figure 2.8. The SOA-based management NGOSS is composed of three different layers: the Application Layer, the Middleware Layer and the Infrastructure Layer. In the first one, the business services such as the B2B integration and business process choreography are located. The second one is an Enterprise Service Bus (ESB), which takes care of providing intelligent routing, protocol mediation and all other communication services internal to the NGOSS. Finally, the Infrastructure layer provides the functions in order to perform Fault, Configuration, Accounting, Performance and Security (FCAPS) management and other maintenance functions. Thanks to the SOA features, in the Application Layer the SID/CIM models can be considered for the creation of directory services of WSDL-described SOA components. The Middleware Layer allows the business processes and services to operate without the direct support of the IT infrastructure, which is reached through the ESB services. It provides integration facilities built on top of industrial SOA technologies and standards, such as XML, SOAP, WSDL, WSAddressing, and WS-Security. It offers a communication channel mostly asynchronous (using the Publish/Subscribe paradigm), which exposes a range of interfaces that are independent from the underlying infrastructure technologies, which is crucial when using heterogeneous security systems and other QoS management functions such as quality measurement, tracing, data management, caching or failure detection and recovery. In this way, different infrastructures can be managed by the application layer in a seamless way, whether ATM, MPLS/IP, FR, Ethernet or other transport technologies are involved.

Beyond the initiatives of standardization bodies, also the research is investigating on the development and deployment of OSS and BSS that are flexible and easy to integrate. In [55], a SOA-based information management model has been proposed. This model is based on the cooperation between several entities, which are wrapped up and implemented as Web Services. The ESB represents the communication layer, which is implemented with the open source Apache ServiceMix software: it combines the SOA functionalities with an Event Driven Architecture (EDA) in order to realize an agile ESB [56]. Other main functionalities of Apache ServiceMix are the support of asynchronous communication and the capability to combine component and services in a vendor independent way. The SOA concepts applied to this information management model allow to ``plug'' in the ESB different FCAPS solutions to maximize and customize the management capabilities.

○ *Considerations about the SOA in Management Technologies*

These works have highlighted that the SOA philosophy introduces the following important advantages in the network management field:

- The crucial management operations can be applied as services: retrieve the status of a device, controlling and changing its configuration settings and provisioning;
- The services are software modules with well-defined, request-response and message-based interfaces, which hide the logic from the users;
- Programming an agent or a management application is a easier job, due to the high reusability of services and the management functionalities that are exposed via consistent interfaces;
- A complex service can perform complex diagnostics requiring the correlation of information from multiple sensors and internal event logs;
- Services can cooperate to provide more sophisticated services constructing more layers on the top of lower-level services. For example, a vendor can provide management agents for an entire range of routers from low-end to high-end, by choosing appropriate services from a service library;
- With advanced SOA implementations, the agents can load new services and protocol adapters without the need to shut down or reboot computers.

2.4 A proposal to use SOA in the Transport Stratum

Next Generation Networks (NGN) are expected to fulfill the custom-need- driven model, which is mainly intended to: satisfy customers in differentiated ways, quickly deploy new services at user request, manage network and service resources in an integrate way. Accordingly, the ITU-T NGN architecture is characterized by the separation of the service stratum from the transport stratum so as to enable: the flexibility to add, maintain and remove services without any impact on the transport layer; the flexibility to add, maintain and remove transport technologies without any impact on the access to service, application, content and information; and finally the optimized usage of multiple access and core transport technologies to form end-to-end connectivity across multiple terminals, different access technologies and different core transport technologies [57]. The service stratum is then responsible for enabling the creation and the delivery of service, application, content and information. As to the transport stratum, it is intended to provide end-to-end connectivity according to the service requirements differentiating the service treatment, the terminal capability and status of the network resource availability.

Differentiation of services in terms of Quality of Service (QoS) is then a must for the transport stratum, which is now possible thanks to the technological advancements achieved in the decade that allow for differentiating traffic delivery in the access and the backbone networks. In a similar way, other works have brought to protocols and technologies purpose-built to support Traffic Engineering operations (TE): they permit to have the optimal traffic distribution avoiding bottlenecks, with consequent link congestion and data loss. The union of TE and traffic delivery differentiation has brought to the development of adequate tools for QoS management, so that network operators have a more efficient alternative than the classic (and more expensive) network over-provisioning. Technology is then not an issue when differentiating services at flow or trunk granularity as well as when engineering the network resources and traffic.

However, a big challenge remains in this context, which is related to the flexibility and dynamicity in the provisioning of new transport services when requested by the service stratum so as to follow the rapid evolution of the user needs and market trends, which is one of the main features of the NGNs. Herein, dynamicity is intended as the capability of the network to bind on-demand services and relevant resources, often provided by different operators through separate domains, at user request and according to his profile. Indeed, the creation of new market-driven applications by reusing an extensible set of existing service components has been a key aspect of telecom platforms for years, but this has almost exclusively characterized the application/service stratum. In truth, application provisioning is currently carried out without the cooperation of the lower layers in the transport stratum, which are responsible for the delivery of the data between distributed service entry-points. Contrarily, to meet the previously mentioned challenge of dynamically providing on-demand services, there is the need for transport functionalities respond to the following demand: provide the upper layers with interoperable and open interfaces so that dynamic binding of distributed application services is synchronized with dynamic activation, configuration and monitoring of transport services.

Our proposal goes in this direction, with the final aim to make available in a flexible and dynamic way all the capabilities of the transport stratum to the upper layers (i.e. the service stratum), to the end-user functions and to other networks. The addressed challenge is the analysis of the transport functional layer as required in the NGN transport stratum, the decomposition of this layer into a set of micro-functionalities according to the Service Oriented Architecture (SOA) style and the implementation of the relevant services using the Web Service (WS) technologies to analyze the advantages of the proposed approach. With the intention to follow the evolutionary approach towards the transition into the NGN networks from the current Internet, this study has been conducted by taking into account the efforts that have been already devoted in the last decade towards the definition of the technologies and protocols to build multi service, QoS-aware and TE-oriented network solutions. In particular, we considered the DiffServ-aware Traffic Engineering (DS-TE) solution, which represents one of the most advanced technologies that performs TE in a differentiated service environment by applying routing constrains with class granularity [58].

This section is organized in the following subsections, which reflect the logical step of this work: the transport stratum architectural model definition, the SOA-oriented design, the system evaluation and the conclusions.

2.4.1 Transport stratum architectural model

Objective of our study is the definition of a SOA-oriented architecture for the transport stratum that should make easy and flexible the provisioning and management of the relevant services, in accordance with requests of the service stratum so as to follow the rapid evolution of the user needs and market trends, which is one of the main features of the NGNs. To this end, we have preliminarily defined a simple but effective layered model for this stratum.

We begin by observing that the deployment of new transport services, starting from a new idea or a new need, implies a sequence of actions and processes that span from the service design to the network devices configuration. Each of these actions belongs to a specific domain of knowledge and implies different processes, which in our scenario should be defined in term of services. Starting from the technical operations and ending with the deployment of complex services, the modeling activity proceeds defining different layers, each one with an higher level of abstraction.

The main functionalities of each layer can be associated to different objects and those who share similar functionalities can be grouped together in order to create new independent components. This

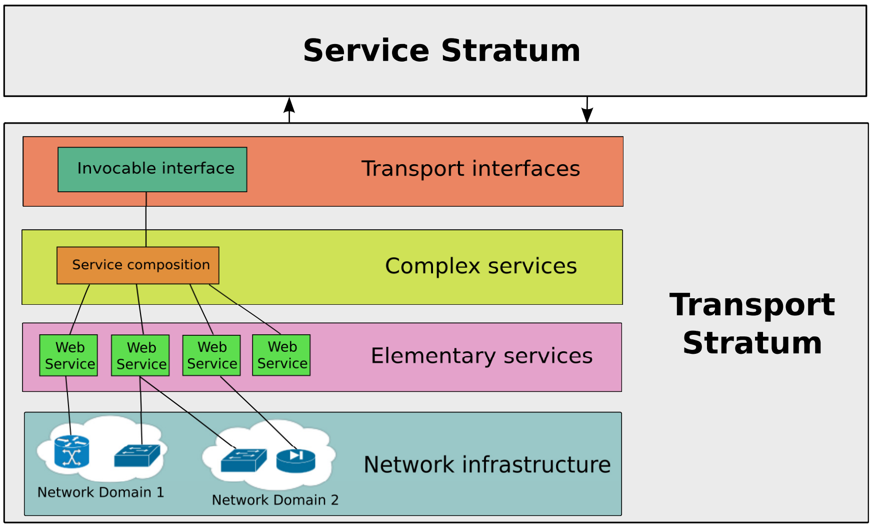

Figure 2.9. SOA-oriented reference architecture for the transport stratum with 4 layers

approach starts from the basic services to the most complex: the services at the higher levels can be obtained as a composition of other services in the lower layers. These principles and considerations are the basis of the proposed SOA-layer architecture for the management and provisioning of the transport stratum, which is shown in Fig.2.9. With reference to the TINA architecture described in the Chapter 1, in this architecture the classical hard-set layered structure of TINA is substituted with a more flexible one, where the creation of complex services is dynamical and flexible, so that new services can be added or modified at runtime, taking advantage of all SOA principles. Even if the proposed architecture implements a service-oriented view too, it makes use of services which are not constrained to communicate onlywith services on adjacent layers, New services can be added and invoked at runtime, and due to fact that are software services, they can be recombined with others. Each service exposes its own capabilities, functionalities and the invocation logistics using a standard and machine-intelligible language.all these information is published to permit the dynamic discovery. The service creation process does not follow a rigid and static structure but it is performed ad composition of elementary services following ad-hoc workflow definitions.

The resulting architecture is made of four different layers. The bottom layer is composed by the physical infrastructure: the devices can belong to a single or to different network domains and regardless of this condition they participate to specific processes defined in the upper layers. The second layer is the Elementary services one; here, the services have the typical properties of SOA services. They are characterized by the features of being independent, self-contained, weakly coupled and reusable, so that more elementary services correspond to a specific process implemented by the transport stratum. Services provided at this layers are: the setting of new transport connections, the monitoring of nodes status, definition of the bandwidth model and setting of relevant bandwidth constraints, configuration of the routing parameters, computation of a new path, just to cite a fews.

In the third layer, specific and complex services are provided, which are composed of more than one elementary service. Using the SOA orchestration principles and technologies, elementary units are connect each other to implement complex services by using a high level of abstraction, which makes easy the service construction process. Their behavior as well as the logic that drives them, can be expressed using workflows of coordinated services. Example of a complex service is the activation of a traffic delivery service in a single domain with QoS guaranteed, which involves the coordinated activation of a series of elementary services, which includes at least: the analysis of the required QoS, the computation of a path satisfying the requirements, the setting of the connection and the allocation of the bandwidth.

On top of the proposed model, the interfaces towards the service stratum are located. Each complex service exposes its own interface, which in turn shows methods to access transport stratum resources. In this way, it is possible to deploy vertical solutions with detailed functionalities. When constructing this architecture particular attention has to be given to the definition of the interfaces at each layer: each interface has to be standardized, so that the complex service as well as the elementary service realization are not constrained to a specific implementation. Using common interfaces, operators without any knowledge of the inner details of each single service, can compose new services tailored to their needs. For example, web programmers can deliver applications that make use of networks devices, without boring with the protocols or primitives for the communication to the devices.

Starting from these principles, in our work we have developed a framework for the construction of such an architecture. For the practical implementation of these concepts, we have chosen to use the WS as well as standard Internet technologies to access to the service provided by each of these layers, except for the network infrastructure. The networking apparati are reached by making use of a layer of specific drivers included in the elementary service implementation as specified in the following. Note that, as shown in Fig. 2.9, there are elementary services that need to connect to the networking infrastructure and others that do not. To the first category belong services that do not involve the transmission of bits or the setting of transport equipment parameters.

- ***SOA- oriented design***

In the design of the transport functionalities according to the proposed model, the crucial activity is the definition of the elementary services to be provided in the second layer. Indeed, the Complex Services layer is where elementary services are combined to provide the transport services, which are then exposed through the Transport Interfaces layer. This process is carried out making use of typical SOA orchestration technologies and no specific considerations are needed for our scenario. Differently, elementary services need to be defined for our specific scenario and, as already stated, these should be independent, self-contained, weakly coupled and reusable; additionally, these should possibly cover all the functionalities expected by the transport stratum, which however may slightly change from a network to another. To this,

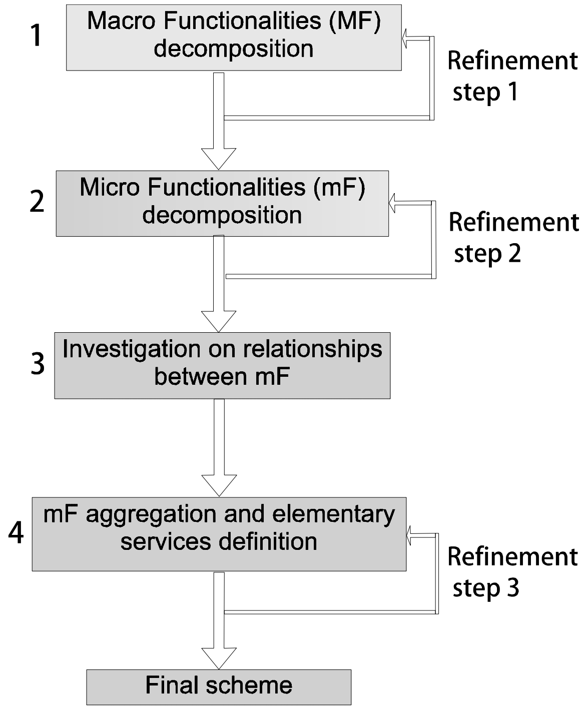
1
Macro Functionalities (MF) decomposition
Refinement step 1
2
Micro Functionalities (mF) decomposition
Refinement step 2
3
Investigation on relationships between mF
4
mF aggregation and elementary services definition
Refinement step 3
Final scheme

Figure 2.10. Steps of SOA-based decomposition

we have analyzed the functionalities of the transport stratum as required in the ITU-NGN architecture and devised the decomposition of this stratum into a set of macro-functionalities and micro-functionalities according to the SOA paradigm. This decomposition has been carried out taking into account the DS-TE network solution. However, the resulting micro-functionalities can be easily generalized for other multi service QoS-aware network solutions. From here we use the acronym ES to refer to elementary service for clarity of presentation.

- ***Transport stratum decomposition***

Herein, we propose a SOA-oriented design applicable to the transport functionalities of multi service telecommunication networks. We take into account the DS-TE architecture, but as already stated, the resulting analysis is quite generic to be applicable to other architectures compliant with the transport stratum functions defined by ITU-T [57]. To decompose the monolithic architecture of a multi service network towards a new SOA-oriented architecture, we have followed the procedure shown in Fig.2.10.

In step 1, the DS-TE architecture is decomposed in macro-functionalities (MF), defined as macroscopic components of the system that take care of well-defined aspects. In our analysis we have isolated the following MFs, which characterize the transport management in NGNs [17]:

- Path computing: it calculates the physical path over the links and devices of the network

- Class Type definition: it defines a set of traffic trunks with similar QoS and bandwidth constraint requirements
- Load Balancing / Performance Management: it focuses on traffic distribution
- Creation and management of bandwidth constraints: it handles the constraints on the bandwidth adopted to achieve the QoS targets
- Resilience / Fault management: with the aim to grant the defined performances in terms of QoS, it recovers network faults
- TE-Routing: together with the Load Balancing funtion, it aims at optimizing the utilization of the network resources
- SLA Management: it handles the customer needs
- Accounting / Admission Control: it authorizes the user connectivity and stores its data for billing
- Policy Management: it manages every policy as defined by the network owner or administrator
- Security management: with the accounting, it gives the rights to use the network and the single user traffic isolation
- Inter-operability with other domains: it manages the connections and traffic delivery between different domains

The main macro-functionalities of the reference architecture are included in this list, but it is not intended to be exhaustive. This set of functions comes from the work done by the ITU-T, but another list could also be considered. However, the intent of this analysis is not the definition of a complete list but the application of the SOA paradigm to the transport stratum, which brings to a set of services that can be restricted, extended and modified without the any need for a complete redesign. The refinement step 1 is driven by the analysis of the DS-TE requirements, as defined in literature [58] [59]. The result of this analysis is a set of MFs that apply to a generic multi service transport stratum.

In the second step every MF, in turn, is decomposed in micro-Functionalities (mFs): each of these represents an elementary task, which can be considered as the smallest part of a generic service. For example, the Path Computation is decomposed in the following mFs:

- Edge routers specification: it defines the traffic ingress and egress devices
- Label definition: it specifies the labels that are used in each link
- Label distribution: it delivers the labels to each specific device

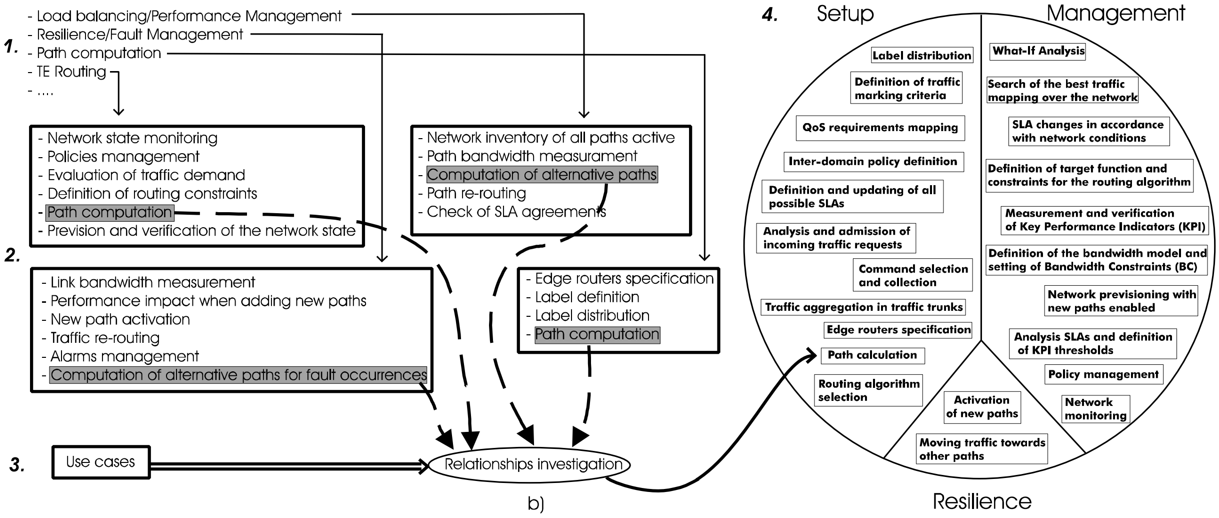
1.
- Load balancing/Performance Management
- Resilience/Fault Management
- Path computation
- TE Routing
- ....
2.
- Network state monitoring
- Policies management
- Evaluation of traffic demand
- Definition of routing constraints
- Path computation
- Prevision and verification of the network state
- Network inventory of all paths active
- Path bandwidth measurament
- Computation of alternative paths
- Path re-routing
- Check of SLA agreements
- Link bandwidth measurement
- Performance impact when adding new paths
- New path activation
- Traffic re-routing
- Alarms management
- Computation of alternative paths for fault occurrences
- Edge routers specification
- Label definition
- Label distribution
- Path computation
3.
Use cases
Relationships investigation
b)
4.
Setup
Management
Resilience
Label distribution
Definition of traffic marking criteria
QoS requirements mapping
Inter-domain policy definition
Definition and updating of all possible SLAs
Analysis and admission of incoming traffic requests
Command selection and collection
Traffic aggregation in traffic trunks
Edge routers specification
Path calculation
Routing algorithm selection
What-If Analysis
Search of the best traffic mapping over the network
SLA changes in accordance with network conditions
Definition of target function and constraints for the routing algorithm
Measurement and verification of Key Performance Indicators (KPI)
Definition of the bandwidth model and setting of Bandwidth Constraints (BC)
Network previsioning with new paths enabled
Analysis SLAs and definition of KPI thresholds
Policy management
Network monitoring
Activation of new paths
Moving traffic towards other paths

Figure 2.11. An example of Web Services functioning and correspondent SOA technologies used

- Path computation: it computes the logical path to be setup

A refinement step is also required in this step, with the aim of having a better knowledge of all tasks participating to the related MF.

The third step concerns the analysis of the interactions between mFs, with the aim of eliminating possible duplications. The global view of all mFs and relationships allows us to see which of these are strongly coupled: this is a key-indicator about a possible aggregation.

In the fourth step, the union of more than one mF runs to the definition of an ES according to the SOA paradigm: loosely coupled, self-contained, interoperable, modular and location-transparent. Also in this phase, the refinement step is fundamental to verify the completeness and the compliance of the proposed solution to the SOA principles. To clarify the analysis we have conducted for every step sketched in Fig.2.10; in the left side of Fig.2.11 we show the procedure we applied to a part of the MFs (four) and relevant mFs. Specifically, on the basis of the scope and objectives of the considered MFs, we have reached the following decomposition (the meaning of each mF is clearly explained by the title itself):

Load balancing/Performance Management:

- Network inventory of all active paths;
- Path bandwidth measurement;
- Computation of alternative paths;
- Path re-routing;
- Check of SLA agreements.

Resilience/Fault Management:

- Link bandwidth measurement;

- Computation of alternative paths for fault occurrences;
- Performance impact when adding new paths;
- New path activation;
- Traffic re-routing;
- Alarms management.

Path Computation:

- Edge router specification;
- Label definition;
- Label distribution;
- Path computation.

TE Routing:

- Network state monitoring;
- Policies management;
- Evaluation of traffic demand;
- Definition of routing constraints;
- Path computation;
- Prevision and verification of the network state.

It can be noted that all these MFs require a mF with the objective to compute the path over the network from two edge routers, for both primary and backup traffic flowing, as highlighted in step 2 of Fig.2.11. This leads to the abstraction and the definition of the Path calculation ES included in the complete list of service stratum on the right side of Fig.2.11.

We also defined 5 use-cases and corresponding workflows: those represent a well-defined sequences of operations that involve more services with the aim to build a complex application. The definition of several use-cases allowed us to improve the mFs definition and interaction. A use-case is defined as a specific management or control procedure of the transport functionalities, which may or may not correspond to one of the initial MFs. In our analysis we have defined 5 use-cases and corresponding workflows: 1) Activation of a traffic delivery service in a single domain with QoS guaranteed, 2) Activation of a traffic delivery service in a multi-domain environment, 3) Optimization of network resource usage, 4) Setup of a bandwidth model over the network, 5) Change of policy QoS for incoming requests. The ESs involved in each use- case and the correspondent WS are shown in Table II, where the right column shows the name we have assigned to the implemented WS. On the right side, Fig.2.11 shows the final and complete list of ESs, that we obtained applying the process of Fig.2.10 to the initial list of MFs. These have been finally grouped in the following three macro-areas: setup, resilience and management.

Table II: Elementary services and correspondent WS involved in each use-case

Use case	Elementary service	Correspondent WS
Use case 1	Traffic aggregation in traffic trunks Path Calculation Command selection and collection Measurement and verification of KPI	Traffic Mapper LSPPathCreation CommandSending KPIValidation
Use-case 2	Inter-domain policy definition All ES defined for use-case 1	InterDomainPolicyDefinition All WS defined for use-case 1
Use-case 3	Network Monitoring Definition of a target function and constraints for the routing algorithm Search of the best traffic mapping over the network Moving traffic towards other paths Measurement and verification of KPI	NetworkMonitoring Constraints Definition TrafficBestMapping TrafficMoving KPIValidation
Use-case 4	Network previsioning with new paths enabled QoS requirements mappings What-if analysis Definition of the bandwidth model and setting of Bandwidth Constraints (BC) Proxy, Measurement and verification of KPI	NetworkPrevisioning QoSMapping WhatIfAnalysis Bandwidth Constraint Settings Proxy, KPIValidation
Use-case 5	Inter-domain policy definition Policy management All ES defined for use-case 4	InterDomainPolicy Definition Policy Management All WS defined for use-case 4

- ***Architecture of elementary services***

The ESs are implemented according to an architecture made of three layers (see Fig.2.12): the Interface Layer, the Business Logic Layer and the Devices Communication Layer.

The Interface Layer exposes the methods available through a standard WS interface. It is responsible for the management of all the incoming as well as outcoming messaging operations involved in the communication with other ESs according to the logic of the complex services in the upper layer.

The Business Logic Layer encompasses the code for the realization of the logic behind the methods implementation. Each method of the web interface is translated into a list of specific procedures that are sent to the devices using the device layer communication's primitives. This layer implements the logic of the elementary services, according to the SOA rules. The implementation of a ES that does not need to communicate with the physical networking infrastructure only relies in the Interface Layer and the Business Layer. Examples of this type of ES are path calculation and QoS requirements mapping. Instead, the WSs which also need to control and manage real devices has a third layer, the Device Communication Layer. A WS with all three layers represents a WS Proxy (WP). Examples of these type of ES are label distribution and network monitoring. The Device Communication Layer is the component used to translate the interface methods into specific device commands. It offers a complete set of primitives that map all the functionalities of the device. This layer is also responsible for the communication session management. It opens a communication socket with the device console and sends all the commands to it using different communication protocols. This layer is the only that needs to change when changing the networking apparatus. Accordingly, adding new types of equipment requires to just add the specific driver that allows for the handling the communication with this new category of devices.

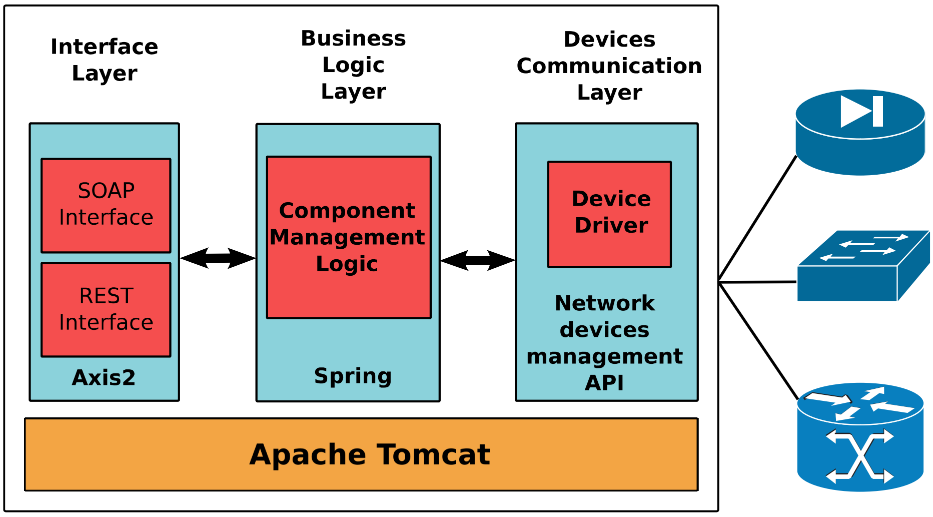

Figure 2.12. Implementation of SOA Ess with WS technology

To easily access the devices functionalities, an object-oriented structure to categorize the network apparatus has been developed. The root of this structure is the generic device class, whereas two different generic classes, the GenericSwitch and GenericRouter inherit its functionalities. In turn, more specific classes related to the vendor and to the specific apparatus can be found. Following the objected-oriented structure of this model, all network apparatus can be categorized by their functionalities as well as their brand, so that it is possible to represent different devices using the same generic classes and interfaces; at the same time, the specific functionalities of each device are implemented with methods which belong to the specific device driver. The inner details of the WS and its functional description can be found in [51].

2.4.2 System evaluation

In this section, we report on our efforts for the creation and evaluation of a working implementation of the proposed system.

- ***System setup***

As already discussed and shown in Fig.2.12, every elementary service is composed at least by two main components that are the Interface and the Business Logic layers, and can also include the Device Communication layer. These have been implemented using the open source framework Axis2 and Tomcat from the Apache Software Foundation, and the software Spring. The business logic layer is constituted by all the code for the realization of the logic behind the methods implementation.

Each method published by the web interface is translated into a list of specific procedures that are implemented using the Java language. The services communicate with the physical devices thanks to the Device Communication Layer: this one offers a complete set of primitives that map the most important functionalities of the device. All WS are deployed and run over an application server, which allows for reaching each WS by means of the HTTP protocol. The logic behind the creation and the management of complex services belonging to the Complex services layer can be expressed in terms of workflows of business processes. A workflow can be represented using a standard language. According to our philosophy to follow the SOA requirements, we have chosen the BPEL (Business Process Execution Language) language. It defines business processes using an XML based language but does not define a graphical representation of processes or provide any particular design methodology for processes.

Finally, each business process has inputs, method and outputs and so, also the workflow exports a web-service interface, which belongs to the transport interfaces layer. We have implemented all the elementary services listed in Fig.2.11 and conducted experiments specifically with reference to the following elementary services and the corresponding WS: TrafficMapper, LSPPathCreation, NetworkPrevisioning, CommandSending, KPIValidation and the Proxy.

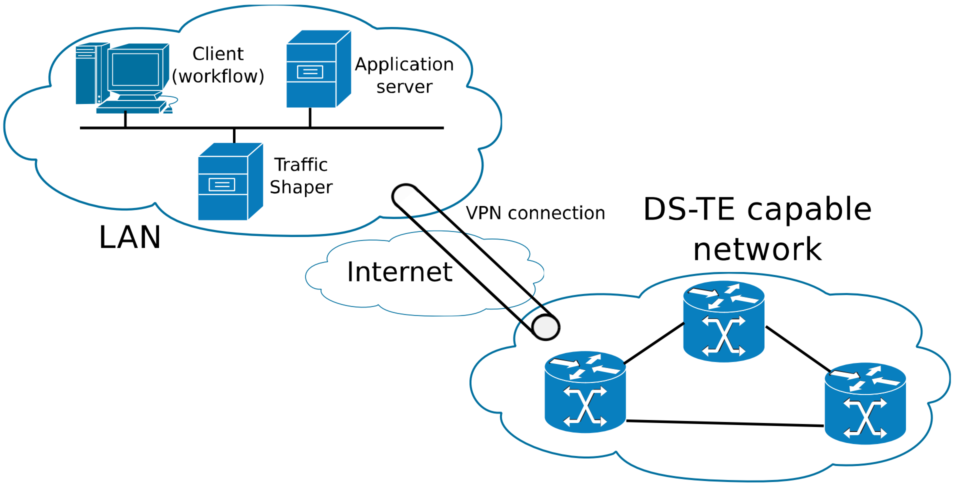

Figure 2.13. An example of Web Services functioning and correspondent SOA technologies used

- ***Experiments***

To evaluate the limits and potentialities of the proposed approach, we have set up a simple test architecture connecting a workstation running a Remote Procedure Call (RPC) client and a server running all the WSs. Besides, thanks to a Virtual Private Network connection, this architecture can reach a DS-TE network hosted in the ISP Tiscali test lab, as shown in Fig.2.13. Herein, we refer to the experiments that have been performed with use-case 1 of Table 1, which implements the activation of a traffic delivery service in a single domain with guaranteed QoS, and whose workflow is shown in Fig.2.14. Similar results have been obtained with the other use-cases. Specifically, in these experiments we wanted to analyze the impact of implementing a complex service with distributed elementary services instead of a monolithic system implementation. With this simple architecture, two test scenarios have been conducted.

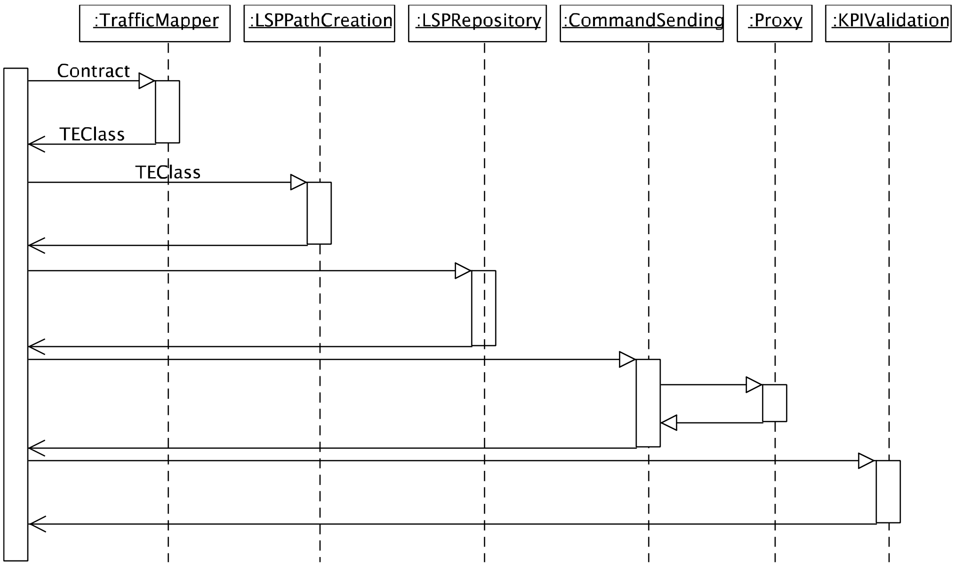

Figure 2.14. Workflow implementation of use-case 1 in Table II (UML sequence diagram)

The first scenario is aimed at analyzing only the performance of the elementary services composition process, without involving real devices so that this test does not include the use of any WP. It means that in the workflow shown in Fig.2.14 we have skipped the call of the Proxy WS. Following the SOA style, our system must be composed only by stateless services. Anyway, a network is not completely stateless: the creation of LSPs or parameters setting determines some different states, which need to be stored. Accordingly, the creation of a new service into a working network cannot neglect the current network state. In our implementation, we introduced the LSPRepository service, which is interrogated and refreshed during the LSP creation; in addition, we also introduced a Network-topology and a Network-link-usage repository in our architecture. We don't create any other service that collects data from physical devices, e.g. using SNMP traps [60]. The connection rate is controlled using a traffic shaper running in another server to simulate network traffic load; all the management traffic is tagged and differentiated from the data traffic, so that it is not affected by congestion and excessive packet delays. The aim of this test architecture is to study the execution time requested by the workflow shown in Fig.2.14, at different WS aggregation levels. The aggregation level specifies how many elementary services, and so their correspondent WS, are combined together inside the same WS.

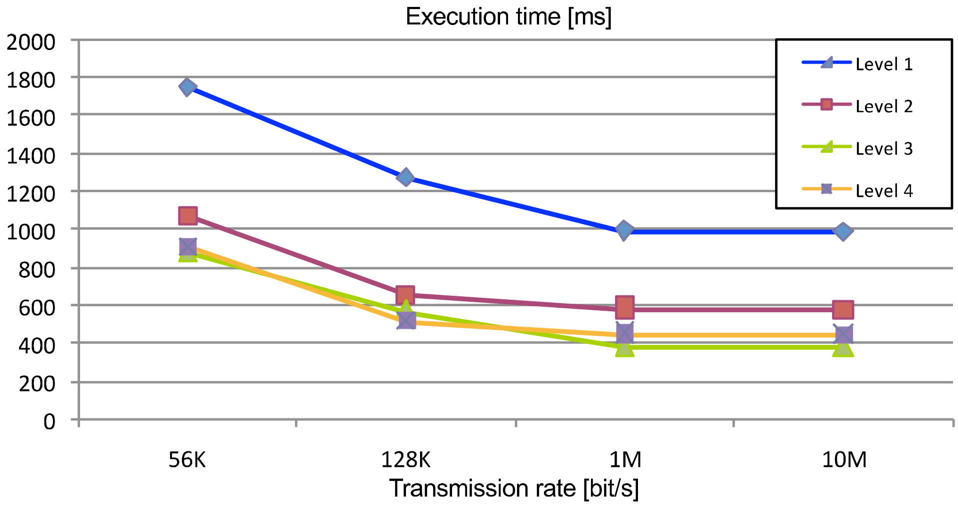

Figure 2.15. Execution time of the workflow of Fig 2.14 with different WS aggregation levels

Reducing the aggregation level, we may distribute the single elementary service in different servers, giving the administrator the flexibility to better distribute the load among several servers according to the traffic generated by each elementary service.

With regard to Figs 2.14 and 2.15, with an aggregation level 1, the client calls each different elementary WSs by sending a different SOAP message request to the server. With the aggregation level 2, the client calls two generic WSs running into the server: each one of these, in turn, call two elementary

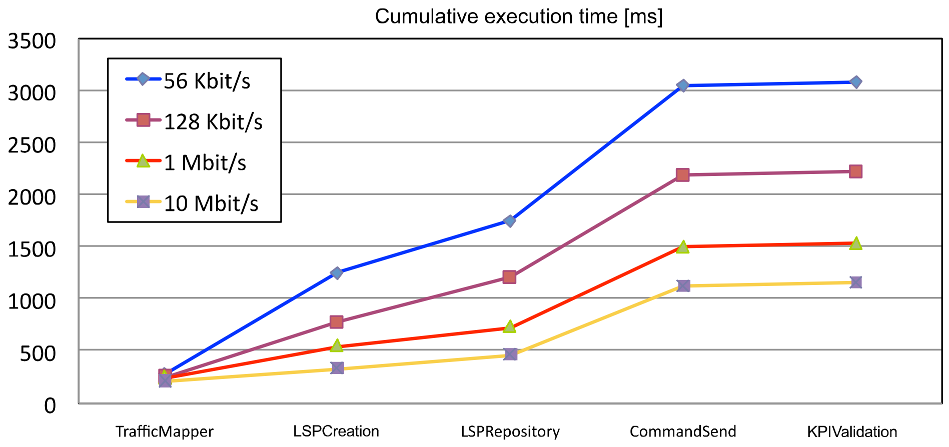

Figure 2.16. Cumulative response time of workflow shown in Fig 2.14.

WSs. With the aggregation level 3, the client calls a generic WS, that in turn calls three elementary WSs, and the remaining elementary service. Finally, with the aggregation level 4, the client calls only one generic WS, which makes four calls to elementary services, all running into the same server.

The simulations show a decreasing execution time as the number of WS aggregating level increases, as expected. Indeed, the total execution time is composed by the sum of transmission time and processing time: with low values of bandwidth, also considering that there is only one hop between the client and the server, the transmission time has more importance that the latter. This choice permits not to consider the processing time, which is related to the internal server load and to its internal features. Results show the tradeoff between the WS separation and the WS aggregation. Separation makes possible to have a more flexible architecture, deployable in a distributed system as a geographical network but with the disadvantage of bigger transmission times. Aggregation provides a less flexible but faster system, where the network transmission time has a reduced influence and the system management is concentrated in a single node with a single point-of-failure.

The second test introduces the use of the WS proxy and its final result is the setting of MPLS commands over physical devices. The reference workflow is again the one in Fig.2.14 and this time we also call the Proxy WS to send the commands to the transport infrastructure. In Fig.2.16 the cumulative response time of the workflow execution is shown. The plots have similar shapes, rescaled due to various bitrates allowed over the network by the traffic shaper. It can be clearly noted the high delay introduced by the execution of the Command-Sending WS, which creates the connections to physical devices invoking the WP. Accordingly, the CommandSending service, that in turn calls the WP, essentially needs more time with respect to other WS for two reasons: the former is related to the delay introduced by the Internet, the latter is due to the access and configuration time of physical devices. In the worst case, the total execution time requires roughly 3 seconds.

2.4.3 Experiences in development and deployment and conclusions

In this section we want to provide our experience with regard to the use and the development of complex services in this framework. We experienced the following advantages:

- The design of a complex service starting on the single elementary services is a simple and intuitive practice from both a technician that knows very well the transport technologies and a high-level language developer;
- The time needed to develop and deploy a complex service is reduced with respect to the whole development of ad-hoc services;
- The adoption of standard and easy-understandable interfaces contributes to improve the learnability of the framework;
- A complex service can also be deployed in a distributed environment as in the case of a telecommunication network, considering that every single service is encapsulated into a web service which is easily reachable through standard Internet interfaces;
- Given a complex service in execution, an elementary service can be added in a dynamic way, without affecting the complex service execution flow.
- The adoption of the proposed SOA architecture does not necessarily require to re-implement all the transport services. Most of these would just need to be wrapped by the required WS interfaces, some others may require to be divided in simpler services and some others to be grouped. The most important activities to be carried out is the definition of all the required simple services and their interfaces according to the principle we have mentioned in several parts of the paper.

At the contrary, we experienced the following disadvantages:

- The re-design of a working transport stratum is inevitably a costly and long process;
- The adopted transport technologies, in this specific case the DS-TE, have to be analyzed and decomposed into elementary services. This process is not simple and costless. In addition, the transformation of a monolithic system in a modular system composed by several elementary services is a necessary task to obtain a SOA-oriented system design;
- Even if according to the SOA principles the services are required to be stateless, in a telecommunication network some services storing information about the network state are required. In particular, we had to introduce additional stateful services such as a repository for LSP, a repository for the network topology and another for the network link usage.

Chapter 3. Bandwidth Management in Next Generation Networks

3.1 Introduction

The implementation of complex Next Generation Network (NGN) solutions requires sophisticated Traffic Engineering (TE) functionalities, which allow network operators to overcome some limitations of traditional IP networks in providing differentiated QoS guarantees.

While NGN diffusion is increasing, network management is evolving towards autonomic approaches, which could be a good solution in managing the complexity of NGN networks so as to make realistic the exploitation of its functionalities [61]. Autonomic management allows network nodes to automatically discover their environment, self-configure and automatically update to adapt to changes [62]. In the last years, several important proposals appeared in the literature in this context. Among these, one of the aspects that has been frequently dealt with is the traffic self-optimization (or self-management), and most of the proposed solutions are nature-inspired [63,64]. The objective of traffic self-optimization is to evaluate all or part of the possible paths that satisfy the requirement of the traffic requests and select the best one to route the flow. When implementing an autonomic approach for traffic optimization, without human intervention, the network autonomously computes the best paths so as they evolve to a stable and robust control point.

In this chapter, we present an autonomic approach for NGN networks to improve the efficiency of TE functionalities while limiting control plane complexity at core nodes. The proposed solution keeps centralized explicit routing implementation while it decentralizes the bandwidth management functionalities, which are aimed at driving routing decision. In this scenario, a distributed bandwidth management approach is proposed, which provides for a self-management module in each node of the network.

This chapter is organized as follows. In the next section, the TE problem is described together with autonomic solutions. Section 3.3 illustrates the bandwidth management in DS-TE and section 3.4 describes the proposed architecture and algorithm. Results are provided in section 3.5.

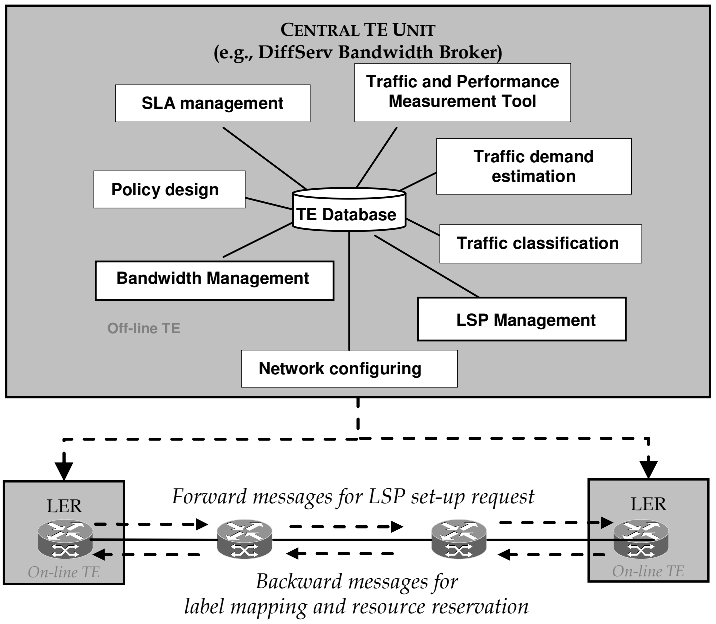
CENTRAL TE UNIT
(e.g., DiffServ Bandwidth Broker)
SLA management
Traffic and Performance Measurement Tool
Traffic demand estimation
Policy design
TE Database
Traffic classification
Bandwidth Management
LSP Management
Off-line TE
Network configuring
LER
On-line TE
Forward messages for LSP set-up request
LER
On-line TE
Backward messages for label mapping and resource reservation

Figure 3.1. The centralized TE architecture

3.2 Autonomic Traffic Engineering

Traffic Engineering [65] is a set of policies and algorithms aimed at guaranteeing the service-specific end-to-end QoS levels at given network resources. This objective is reached by balancing network traffic load so as to minimize congestion risks and improve network resource utilization. Due to their complexity in current NGN networks, TE functionalities are often performed by human administrators or deployed to work off-line in centralized TE units. At the same time, some on-line TE functionalities are performed by edge routers, which manage the control plane functionalities in the network, whereas the core nodes focus on switching operations to reach high forwarding rates. However, this centralized or edge-based approach often results inaccurate since TE networks evolve dynamically and state information quickly became out-of-date. Due to this reason, in the last years TE algorithms and approaches are evolving toward solutions that allow overcoming traditional limitations, spreading TE functionalities on the whole network and involving all network nodes in route computation. This section presents the main component of Traffic Engineering network architecture, describing all functions and roles and the evolution of TE approach.

3.2.1 Traffic Engineering functionalities

Fig. 3.1 shows the main components of a Traffic Engineering unit. It relies on an accurate traffic demand estimation, which can be derived from either the service level agreements (SLA) or the network monitoring tool. Based on the expected traffic load and services, policies for network management are defined to drive the processes of traffic classification, bandwidth allocation and route computation. A central data repository stores all shared network information, such as topology, link state, traffic demands, routes and policies and allows edge routers to perform explicit routing with QoS guarantees. Link state information are constantly updated thanks to the traffic and performance measurement tool. All TE functionalities are usually performed by a remote central unit, i.e. the Bandwidth Broker (BB) of DiffServ architectures, while some limited functionalities, such as Traffic Classification and path management, may be performed also by the LERs (Label Edge Router).

Default TE implementation works on a flow-based approach, where each traffic flow is marked by the LERs (Label Edge Router) with a label corresponding to the forwarding along a specific label switching path (LSP). LSPs are unidirectional: for each pair of LERs, the one that transmits with respect to the direction of data flow is said to be upstream, whereas the LER that receives is downstream. The computation of LSPs is performed through CBR algorithms [65]. They are used by the upstream LER to find an explicit route in the network that meets a set of constraints on QoS and resource utilization, such as bandwidth requirements and availability (e.g., the bandwidth bottleneck in the path), key performance indicators (delay, packet loss ratio) or resource classification. When a new LSP setup request arrives in a TE domain, the downstream LER runs a label distribution protocol to require label mapping and bandwidth reservation to all nodes along the explicit route. There are currently two label distribution protocols that support for TE LSP setup: extensions to RSVP for LSP Tunnels (RSVP-TE) and Constraint-based Routed Label Distribution Protocol (CR-LDP). Both these protocols implement the signaling mechanism shown in Fig.3.1: a label request message is sent from the ingress to the egress node; then, label mapping or reservation messages are sent backward to all LSRs in the path. The setup of a new LSP can require also the preemption of active LSPs with lower preemption priority.

3.2.2 Autonomic Approach

Human centralized management of the complex current NGN solutions is often impossible if we aim at fully exploiting the network potentialities and guaranteeing the required QoS for the heterogeneous set of provided services. Indeed, the network state is always evolving due to continuous setting, rerouting and termination of LSPs, so that information on network state quickly result out-of-date and routing decisions inaccurate. It is then desirable that nodes automatically discover their environment, self-configure and automatically update to adapt to changes.

Autonomic management is a good solution to reduce the complexity of running NGN networks so as to make realistic the exploitation of its functionalities. It mainly consists in self-configuring, self-healing, self-optimizing, and self-protecting. Self-configuring networks provide low cost management and facilitate adaptation to traffic and new services. Self-healing networks reduce the burden of management failures. Self-optimizing networks make efficient use of the underlying components and enhance scalability. Self-protecting networks are robust to denial of service attacks and other external threats [Manousakis et al., 2005]. All these functionalities have to be performed in a distributed way, limiting computational complexity on each single node and giving at the same time good levels of accuracy as in the centralized approach. One of the aspects that has been frequently dealt with by past works is the self-management of the QoS and traffic, and most of the proposed solutions are nature-inspired. A first attempt was proposed in [62], where the authors suggest a nature-inspired framework based on artificial agents that monitor the traffic in a given network and decide routes while network-wide resources are efficiently used. Each agent monitors a routing path for each design class of services. The objective of this nature-inspired algorithm is to achieve long-term high resource utilization and to reduce traffic contentions for various network conditions. In [63], the authors proposed AutoNet, a self-organizing management system for traffic engineering in core networks. They design long-term and short-term control systems to direct the evolution of the network state in the direction of increasing robustness. The long-term control aims to develop the evolution based on an initial knowledge that consists of the business policy as well as empirical results from previous experience about customer demand, network element reliability, price elasticity, etc, while the short-term part reacts to the network changes in real-time. The traffic engineering algorithm chooses the paths in the direction of preserving the robustness of the network to the unforeseen changes in topology and traffic demands. The results are delivered to a General Resource Manager unit which executes orders that allocate the appropriate amount of resources, while random walks continuously monitor the network to identify possible problems (e.g., SLA violations, failure alarms and so on).

3.3 Bandwidth Management in DS-TE

While in previous section we described the general context of TE applications and evolution approaches, in this section we focus on a specific TE functionality in a specific TE architecture, that are the bandwidth management in DS-TE networks. Our interest about this theme is due to two main reasons. The first one, the bandwidth management is a young problem still open. It was introduced in network management when quality of service became a key requirement of network performance; some standards and algorithms have been deployed, but, as described in the following, there are still some limitations which have to be addressed. The second reason is the efficiency of a new TE approach which may results from optimizing just bandwidth management implementation. Indeed, bandwidth management is usually performed before LSP management and, if well performed, may help network nodes to optimize routing and dynamic network management with a limited impact on current implementation and computation load.

DS-TE bandwidth management consists of two main tasks: the choice of a BC Model and the setting of the bandwidth constraints for each active class type. Regardless of which model a network operator decides to adopt, it is required to implement a methodology to configure the bandwidth constraints to efficiently and cost-effectively set up the LSPs belonging to the employed CTs. Notwithstanding the importance of this task, the IETF does not suggest any criteria to be considered. In [66], we proposed a methodology to drive at the same time the configuration of the BC Model and the classification of the traffic into CTs by performing traffic estimation and considering end-to-end quality requirements. This basic approach provides major benefits in the exploitation of network resources while satisfying QoS requirements. However, it can only be implemented off-line by a central entity (i.e. the Bandwidth Broker in the DiffServ architecture) and for this reason it is difficult to quickly react to the varying traffic conditions. To improve the rate of updating and the efficiency in configuration, this solution has been simplified making use of the genetic algorithms [67], which allows LER nodes to set and update bandwidth constraints in on-line mode. However, all control plane intelligence is still confined to edge nodes, which are required to have full knowledge of network status. This may result in out-of-date BC configuration, which brings to erroneous routing decision.

To overcome limitations of these approaches and to improve BC Model efficiency, an autonomic approach is highly demanded, which is able to dynamically self-configure bandwidth constraints based on real network status and bandwidth availability, while it avoids the rise in control-plane complexity of core nodes.

3.4 Distributed Bandwidth Self-Management

In this section we illustrate a bandwidth self-management approach, which defines bandwidth constraints on each link based on local monitoring rather than on traffic demand estimation and full knowledge of network state as in the classical centralized approach. This is aimed at distributing the computational complexity of bandwidth management over all the network nodes and reducing the necessity of continuously transmitting measurement data from/to a central control point. This approach also allows for increasing the reactivity of the system in updating the bandwidth settings based on local traffic changes, while limiting control plane computational load at the core node.

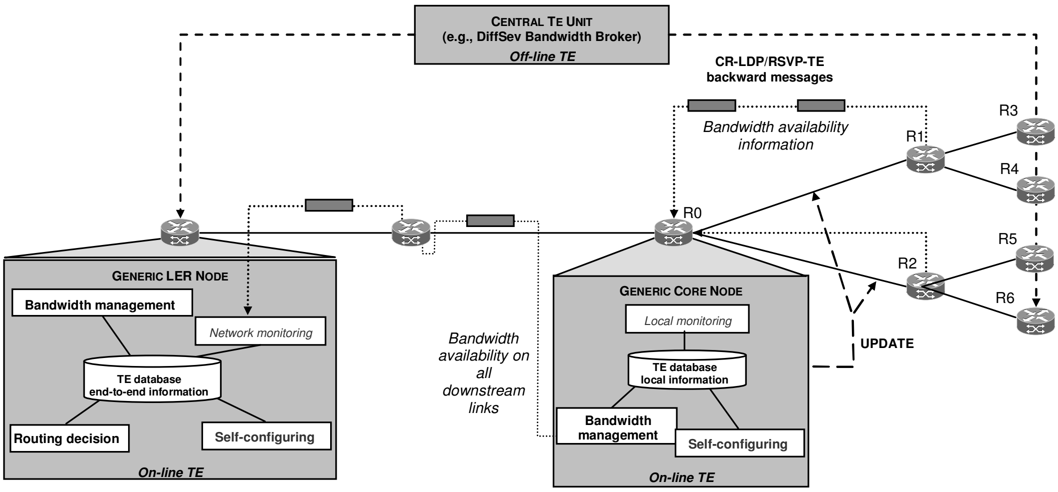

Figure 3.2. Distributed bandwidth self-management architecture

Fig. 3.2 shows our proposed self-management architecture for DS-TE. As it can be noted, there is still a central TE unit that performs some TE settings, such as managing the contracts with the users through SLA subscriptions, performing policy design, defining traffic classification rules, initializing routers configuration and computing off-line routes. On-line routing is still implemented by LERs, while a bandwidth management module is added to each network node for self-configuration. This module works on the basis of local information such as traffic load on each interface, queue status and link operation modes. It also takes into account indications from the central TE unit on the reference BC Model and global policies, such as: the number of active class types, resource classification policies, congestion thresholds, maximum acceptable queue delay per CT. Results of processing allows for setting BC for each CT in each downstream link.

We assume the configuration of the bandwidth constraints is initialized by the TE central unit according to some predictions of the traffic to be served and on the basis of the TE policies (for instance the algorithm in [64] can be used). Then, local bandwidth management modules drive the update actions, starting from the LSRs adjacent to the egress LERs (in Fig. 3.2, R3, R4, R5 and R6) such as the processes of bandwidth reservation and label allocation for LSP setup. Each node examines information on bandwidth availability it receives from its directly connected downstream nodes, computes the new bandwidth configuration for its downstream links and sends the resulting bandwidth availability values to the next node in the upstream direction. This process is repeated by all nodes until the ingress LERs, which collect all information from the core nodes and computes the routes based on up-to-date network state information. Note that, since bandwidth management is performed locally by each node, our proposed self-configuring architecture also enables hop-by-hop implementation of some CBR algorithms, like for example the

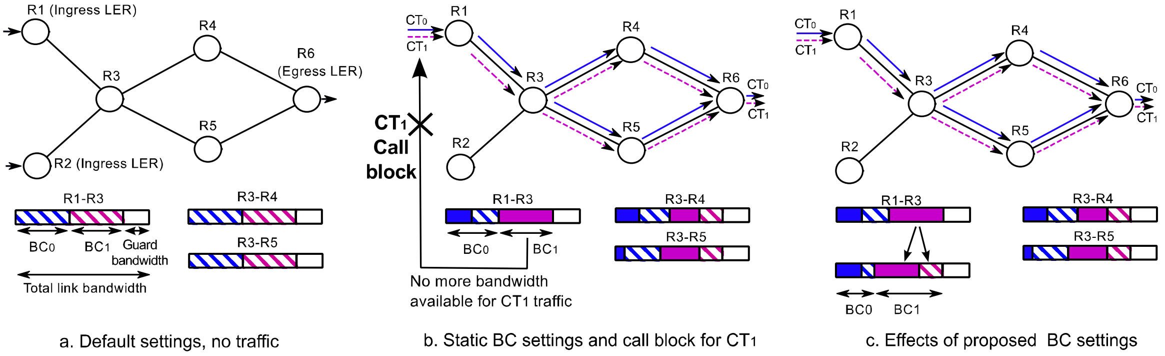

Figure 3.3.Example of BC settings (the dashed bandwidth portions are unreserved bandwidth, the full ones are allocated bandwidth)

shortest-widest path (SWP), which focuses on the bottleneck bandwidth and routes the traffic along the path with the largest unreserved bandwidth [64].

The choice to work in the upstream direction was driven to enhance bandwidth setting, which has to take into account downstream node and link state to avoid congestion or under-utilization. Further, it exploits the capabilities of the stack of protocols used in DS-TE, limiting the complexity of the control plane in the core nodes. Indeed, updates between adjacent nodes may be exchanged using the backward messages of the label distribution protocol, CR-LDP or RSVP-TE, without defining other signaling protocol. In particular, when CR-LDP is used, since it relies on TCP sessions, bandwidth information can be inserted in the Label Mapping messages, which refer to LSP setup, or in the Notification, Label Release, Label Withdraw and Label Abort Request messages, which are used to manage an existing LSP; when RSVP-TE is used, bandwidth information is coded in the RESV messages.

3.4.1 The basic principle

Our proposed bandwidth management module aims to setup and update the bandwidth constraints on each link on the basis of the unreserved bandwidth (UB), which is defined as the difference between the bandwidth constraint and the currently allocated bandwidth. This parameter is one of the main cost metrics taken into account by the CBR algorithms. Based on the measures of the UB, the proposed algorithm aims at smoothing abrupt differences in link bandwidth availability to drive the routing decision towards traffic balancing and then a better utilization of network resources. To explain the basic principle, we refer to a simple scenario we can face in a TE network topology, like the one shown in Fig. 3.3. In this example, the node R3 has to make routing decision to forward the traffic along the two available paths from R1 and R2 to R6. We also suppose to have two active class types (CT0 and CT1) and to set the bandwidth constraints BC0 and BC1 to reserve to each CT 40% of link capacity. Fig. 3.3.a shows the default setting. Note that this is a reasonable configuration since the remaining 20% of bandwidth per link is taken as guard bandwidth to limit the congestion risk. In this scenario, the CBR will balance the traffic to R6

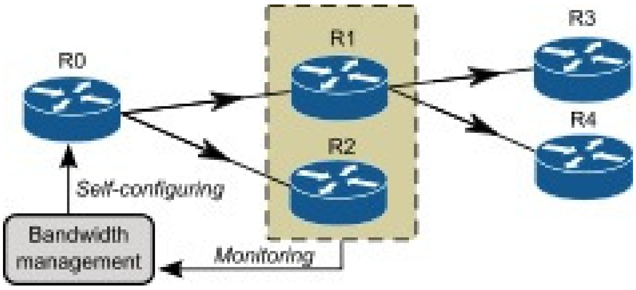

Figure 3.4. The self-management network scenario

between the two available paths until there is available bandwidth in all links in the paths. Now, if we adopt a static BC configuration or a central bandwidth management, which is not able to quickly react to real traffic changes, configuration may bring to the case of Fig. 3.3b, where the link R1-R3 has not more available bandwidth for CT1 traffic so that new CT1 calls are blocked while unreserved bandwidth is also available in the further downstream links. Differently, if we can dynamically and rapidly modify the BC setting so that the BCs on a link are updated based on the status of its downstream links, the network will be still able to route all traffic (Fig. 3.3.c). In particular, we define that the UB on each link has to be equal to the average UB computed in the downstream links. In the considered example, we have assumed the UB of CT1 in links R3-R4 and R3-R5 to equal to 25% and 35% of the total capacity, respectively. The UB of CT1 in link R1-R3 has then been modified to 30% (see Fig. 3.3.c). Accordingly, each node can satisfy the requests its downstream nodes are able to route to destination. In Fig. 3.3.c, we also show as the both CT0 and CT1 have been modified to take into account the guard bandwidth.

3.4.2 The algorithm

The proposed algorithm updates bandwidth constraints in each node at time instants kΔ, with k = 0,1,2,…, where Δ is the update interval. To explain how the proposed algorithm works, we refer to the network scenario in Fig. 3.4 and consider the router R0. It collects information from its downstreams R1 and R2 on the unreserved bandwidth values for each i-th CT and each j-th link among their directly connected downstream links. Then, R0 computes for each of its downstream links and for each CT an optimal unreserved bandwidth value : this is defined as the best value to be adopted for reaching load balancing and it corresponds to the sum of the unreserved bandwidth values received from all downstream nodes, each considered with a different weight. With reference to the example of Fig. 3.4, when computing the optimal UB for the link between R0 and R1, information from both R1 and R2 are considered but with different weights, so that the average UB received from R2 has less influence. To allow the network to quickly react to possible congestion events, the algorithm also ponders the received information according to the status of the links and queues, decreasing the weights of nodes which are experimenting either link failures or high delays. Each optimal value is then compared with the corresponding current value and with the optimal value of the previous update action to evaluate if the evolution of the network is following the desired trend. Based on the results of these comparisons, the new update action is defined. In particular, the algorithm takes into account the worst case, that is the that more differs from the corresponding current and previous optimal values. If the link load is evolving according to the previous update action, just the has to be modified. If the trend is opposite to the previous update action, it is necessary to change the UB of all R0 downstream links. This ensures a faster rate of convergence that is the number of update actions needed to adjust the load balancing.

Each update is driven according to the steps which are shown in Fig. 3.5 and described in the following. We refer to the nomenclature defined in Table III.

Table III . List of variables used by the algorithm

Name	Description
i	CT index
j	Link index
N	Total number of downstream links per node
w	Weight of optimum UBi,j in the j-th link
B_j	Bandwidth of the j-th link
$RB_{i,j}(t)$	Reserved bandwidth for the i-th CT in the j-th link
GB_j	Guard bandwidth for the j-th link
$UB_{i,j}(t)$	UB for the i-th CT in the j-th link
$UB_{i,j}^{f}(t)$	Fractionary value of UB for the i-th CT in the j-th link
$UB_i^m(t)$	Mean of the UB value for the i-th CT
$UB_{i,j}^{opt}(t)$	Optimum UB value for the i-th CT in the j-th link
$\alpha_{i,j}(t)$	Optimal approaching value for the i-th CT in the j-th link

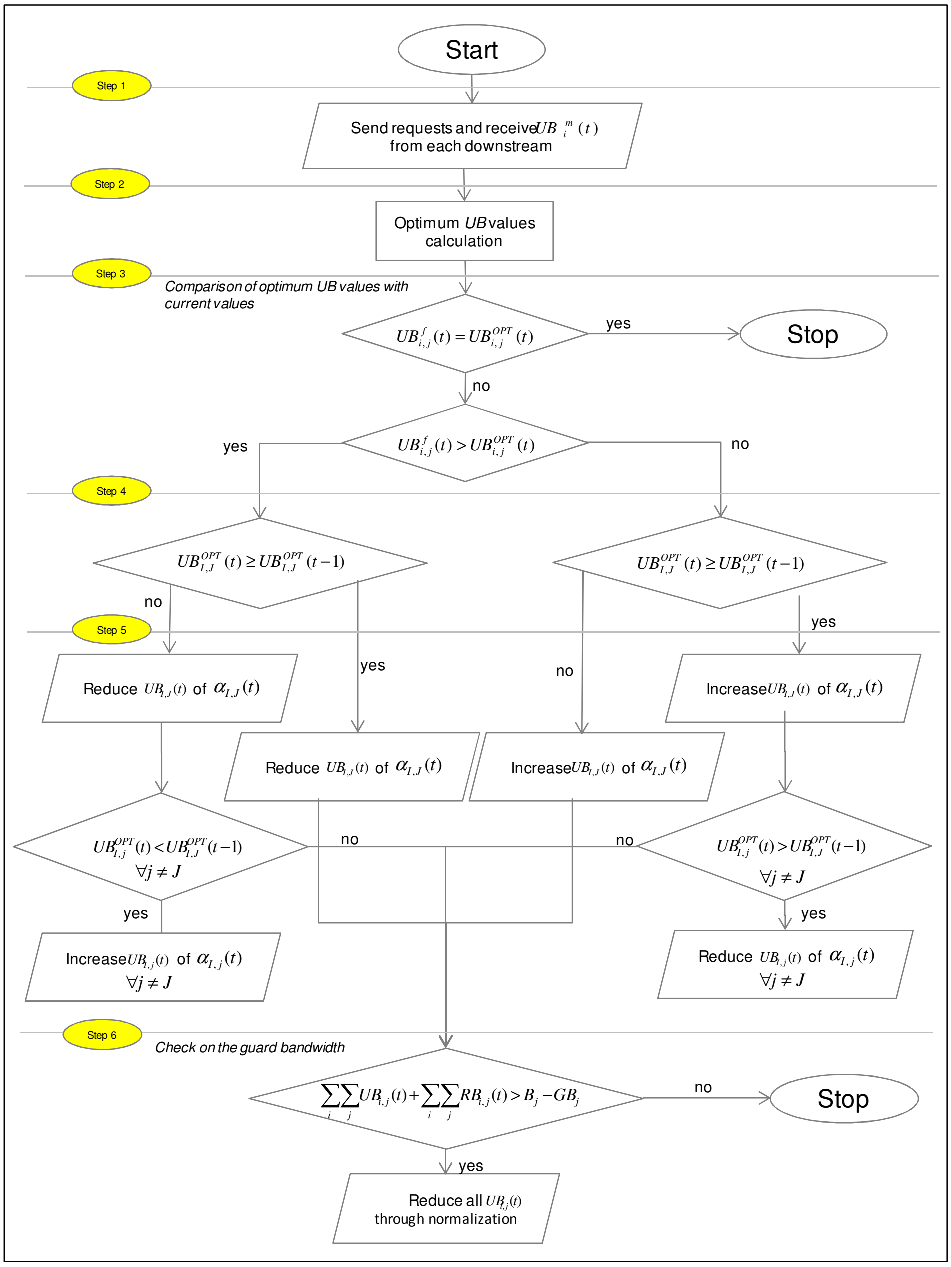
Start
Step 1
Send requests and receive $UB_i^m(t)$ from each downstream
Step 2
Optimum UB values calculation
Step 3
Comparison of optimum UB values with current values
$UB_{i,j}^f(t) = UB_{i,j}^{OPT}(t)$
yes
Stop
no
$UB_{i,j}^f(t) > UB_{i,j}^{OPT}(t)$
yes
no
Step 4
$UB_{I,J}^{OPT}(t) \ge UB_{I,J}^{OPT}(t-1)$
$UB_{I,J}^{OPT}(t) \ge UB_{I,J}^{OPT}(t-1)$
no
yes
Step 5
Reduce $UB_{I,J}(t)$ of $\alpha_{I,J}(t)$
yes
no
Increase $UB_{I,J}(t)$ of $\alpha_{I,J}(t)$
Reduce $UB_{I,J}(t)$ of $\alpha_{I,J}(t)$
Increase $UB_{I,J}(t)$ of $\alpha_{I,J}(t)$
$UB_{i,j}^{OPT}(t) < UB_{I,J}^{OPT}(t-1)$ $\forall j \neq J$
no
no
$UB_{i,j}^{OPT}(t) > UB_{I,J}^{OPT}(t-1)$ $\forall j \neq J$
yes
yes
Increase $UB_{i,j}(t)$ of $\alpha_{I,j}(t)$ $\forall j \neq J$
Reduce $UB_{i,j}(t)$ of $\alpha_{I,j}(t)$ $\forall j \neq J$
Step 6
Check on the guard bandwidth
$\sum_i \sum_j UB_{i,j}(t) + \sum_i \sum_j RB_{i,j}(t) > B_j - GB_j$
no
Stop
yes
Reduce all $UB_{i,j}(t)$ through normalization

Figure 3.5. Flow chart of the proposed algorithm

The algorithm is composed by six steps. In the first step, each node sends requests to its downstream routers and collects the $UB_i^m(t)$ values, computed by each downstream according to the (3.1).

$$UB_i^m(t) = \frac{1}{N}\sum_{j=1}^{N} UB_{i,j}^f \tag{3.1}$$

where

$$UB_{i,j}^f(t) = \frac{UB_{i,j}(t)}{\max BandwidthAvailable} \tag{3.2}$$

In the second step, the optimum values $UB_{i,j}^{opt}(t)$ are computed (see (3.3)): these values are the best values to be adopted for reaching the load balancing in the network.

$$UB_{i,j}^{opt}(t) = w_j \cdot UB_{i,j}^{m}(t) + (1-w_j)\frac{1}{N-1}\cdot\sum_{\substack{k=1\\k\neq j}}^{N} UB_{i,k}^{m}(t) \tag{33.}$$

In the third step, the algorithm compares the values obtained in the previous step with the current $UB_{i,j}^{f}(t)$ set in each node: the biggest difference permits to individuate the CT-link pair (I,J) whose UB setting has to be modified for reducing the difference from its optimal value. So, in the following steps some correction actions are performed. In particular, in the step 4, $UB_{I,J}^{OPT}(t)$: is compared with the optimal value of previous update action ($UB_{I,J}^{OPT}(t-1)$): this allows to evaluate if network evolution is following the desired trend. Then, some different actions can be performed (step 5). The following scenarios are possible:

- $UB_{I,J}^{OPT}(t) \in [UB_{I,J}^{f}(t), UB_{I,J}^{OPT}(t-1)]$: this means that the load is evolving according to the previous update action and only $UB_{I,J}^{f}(t)$ needs to be modified
- $UB_{I,J}^{OPT}(t) \gtrless [UB_{I,J}^{f}(t), UB_{I,J}^{OPT}(t-1)]$: this means that the load is not evolving according to the previous update action and there is then the need for a change of all $UB_{I,J}^{f}(t)$.

In all cases, the $UB_{I,J}^{f}(t)$ values are modified (increased or decreased according to comparison results) by the quantity $\alpha_{i,j}(t)$ that is defined as in the following.

$$\alpha_{i,j}(t) = \frac{\left|UB_{i,j}^{f}(t) - UB_{i,j}^{opt}(t)\right|}{UB_{i,j}^{opt}(t)} \tag{3.4}$$

Finally (step 6), it has to been taken into account the guard bandwidth value: every turn a UB is being modified, the sum of the reserved and the unreserved bandwidths on a link can go over into the guard bandwidth; in this case, all UB values on the link have to be reduced through normalization.

The proposed algorithm has been designed to work with all standard BC models. Specific extensions can be introduced to apply the algorithm also when non-standard models are used in the network. When using the MAM model, we verified that the proposed algorithm converges to a better solution if we impose the independence between BCs defining a maximum BC value for each CT on each link and allowing for having a portion of unreservable bandwidth for one or more CTs. Indeed, simulations of different configurations demonstrated that, if we link the variations of a BC to the others, the traffic tends to converge toward a situation when a BC dominates all the others. As to the RDM model, since this model provides for bandwidth sharing between CTs and interdependence between BCs, we defined the constraint to run the algorithm independently for each BC, starting from the BC corresponding to higher priority

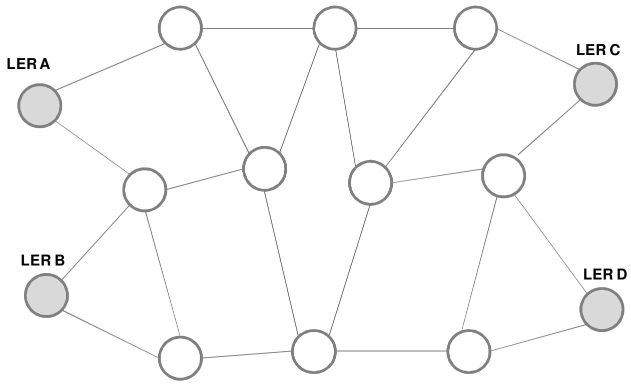

Figure 3.6. Network topology

traffic. Differently from MAM, RDM then requires M-1 update actions at each iteration, where M is the number of active CTs.

3.5 Experimental Results

In this section we show the main performance results of the proposed approach when applied to a simple DS-TE domain with 10 core routers, 4 LERs and links of 155Mbps capacity, whose topology is shown in Fig. 3.6. Three CTs are considered to differentiate the network behaviour for real-time applications (CT2), non-real time applications (CT1) and services which do not require any QoS guarantees (CT0). For each run, we have simulated the requests of K=200 LSP setup belonging to each of the three CTs with equal probability and randomly selecting the pair of edge nodes. The traffic load for each request varies according to a uniform distribution in the range [0, n]% of the link capacity, where n has been varied in the range 0.3-2.5 for different simulation sessions. The interarrival request time follows an exponential distribution with mean time equal to 15 times the update interval Δ.

In Figs. 3.7 and 3.8 we show the benefits of our algorithm against a default BC setting configuration where the BCs are kept fixed to the starting values (each 30% of the link capacity) and n has been set to 1. Fig. 3.7 shows the variance of unreserved bandwidth among all network links over the time. The instants where the behaviour of both the curves varies correspond to the setup of a new LSP. As it can be noted, applying a static BC configuration the variance increases with the number of active LSPs, which means that traffic load balancing progressively moves away from the desired configuration. Differently, the self-configuring system is able to reduce the variance of the UB with a few numbers of update actions after each new setup. Fig. 3.8 shows the average variance of the UB for all CTs during the simulations considering both MAM and RDM models. Six different values of n have been considered.

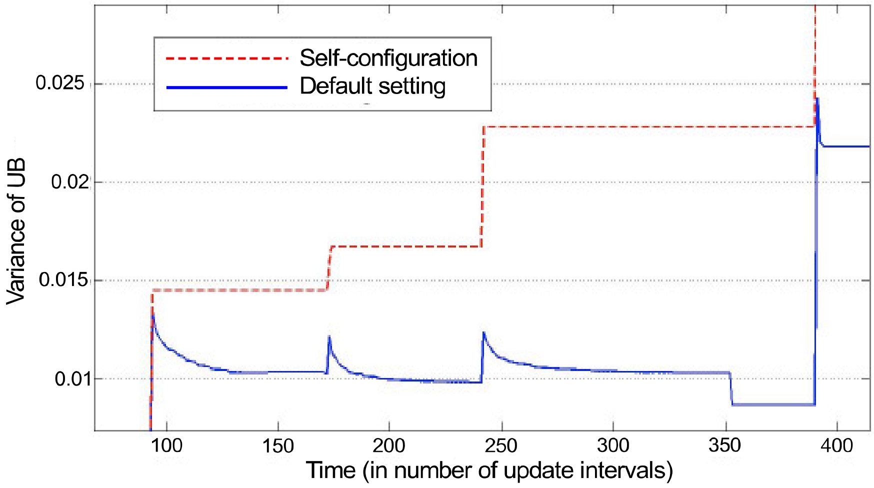

Figure 3.7. Effect of update action on the UB variance over the time

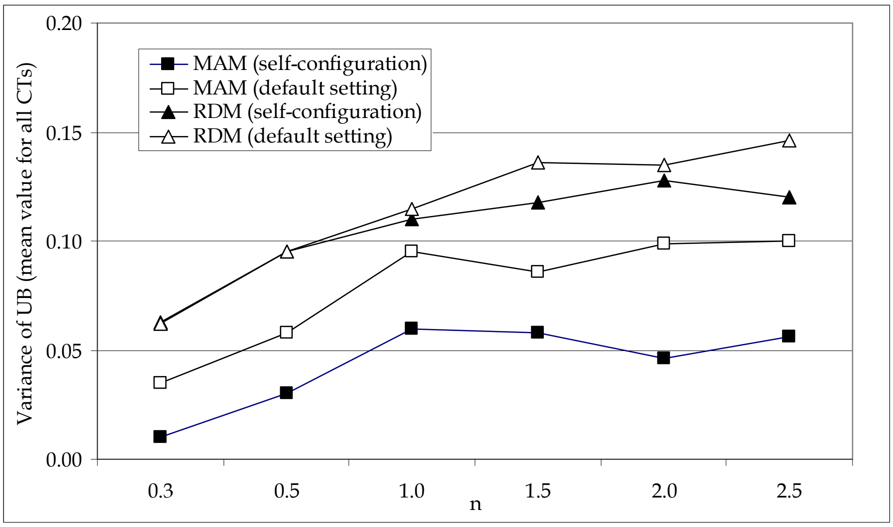

Figure 3.8. Variance of the unreserved bandwidth with MAM and RDM Models (mean

In the case of the MAM model, we can note that our algorithm allows for lower values of variance for all CTs and all traffic load granularities with an average gain of 0.35 over the static setting. In the case of the RDM model, we have not obtained any advantage when the traffic load per LSP is low, whereas we have obtained an average gain of 0.12 when n ≥ 1. Indeed, with high traffic granularity and with static BC setting the CBR is still able to uniformly distribute the traffic over the network so that the drawback shown in the example of Fig. 3.4 happens only rarely.

To demonstrate as the behaviour of the variance of UB influences the network performance, in Table III we report the values of blocking and preemption rates for MAM and RDM models, considering the average values for all runs (0 < n < 2.5) at the end of the simulations. We can note that the blocking rate for all CTs is always higher when using the default BC setting. When using the RDM, the benefits of the proposed strategy increase for higher priority traffic due to the fact that in this case the constraints are adjusted starting from the highest priority traffic and then proceeding with the others, which then obtain lower benefits. From Table 2, we see that the improvement in the number of LSP settings is obtained at the expense of higher numbers of preemptions. This is due to the fact that in general the number of preemptions increases rapidly when the network link occupation is very high, which is the case when adopting the proposed strategy.

Table IV. Performance of the proposed algorithm

	MAM	**RDM**	**MAM**	**RDM**
	Default setting	*Self-configuration*	*Default setting*	*Self-configuration*
Blocking rate CT0	0.5444	0.5351	0.5667	0.5074
Blocking rate CT1	0.6257	0.6156	0.5713	0.5193
Blocking rate CT2	0.6016	0.5963	0.5498	0.428
Preemption ratio	-	-	0.13	0.27

3.6 Conclusions

This work illustrates the problem of bandwidth management in NGN architectures, with specific regard to DS-TE scenario where the setting of the bandwidth constraints is an important parameters of Traffic Engineering approach. Indeed, DS-TE architectures implement TE in a per-class basis making use of bandwidth constraint models, which control the amount of resources assignable to each traffic class per link and influence the execution of the routing algorithms, then representing a powerful tool to drive the utilization of the resources in the network and the fulfilment of the QoS requirements. Starting form this assumption, a new bandwidth management solution is proposed, which relies on a distributed and self management approach. A self-management module is implemented in each node of the network: it monitors the unreserved bandwidth in adjacent nodes and adjust the local bandwidth constraints so as to reduce the differences in the unreserved bandwidth of neighbour nodes. Accordingly, we smooth abrupt differences in bandwidth availability along possible paths, which are frequently due to static settings of bandwidth constraints. Due to the distributed nature of the proposed solution, the adjustments can be frequently introduced, allowing for a quick adaption of the network to the traffic changes.

The proposed solution has been compared with static bandwidth constraint settings in terms of resulting bandwidth blocking rates. The experimental results showed that it allows for obtaining an increase in the number of LSPs that the network is able to accommodate. When adopting this strategy, the network operator has just to set a few weights and leave the network autonomously correct the setting as the traffic and link occupation levels change.

Chapter 4. Peer to Peer Video on Demand Network Architectures

4.1 Introduction

Last years have been characterized by an exponential growth of video traffic on the Internet, which has brought to significant investments in networks and systems aimed at the delivery of real-time high-rate streams. Several traffic analyses tell us that this growth will continue over the next decade, making video streaming applications the ones driving the Internet evolution during the near future [68]. Video on Demand (VoD) is one of these applications, which requires resources able to deliver a video whenever the customer requests it. Realizing a VoD system using the Internet requires architectures tailored to video characteristics. Even if advanced video coding technologies such as Scalable Video Coding (SVC) [69] allow for an efficient representation of the video content towards the transmission over packet networks, VoD service requires guaranteed bandwidths and constrained transmission delays that make it quite difficult to be provided in the traditional Internet architecture.

Typical VoD solutions can be grouped into four categories [70]: centralized, proxy-based, Content Delivery Network (CDN) and Hybrid architectures. In a centralized architecture, the source server manages all clients: it is the simplest and easiest way to implement a VoD system. This solution has the big disadvantages of having a single point of failure, requiring servers with high computational and transmission capabilities that generate unbalanced network loads. Proxy-based architectures are aimed at decreasing the central server load, introducing proxy-servers in strategic points of the network, typically close to the clients. CDNs can be seen as an extension of the proxy-based approach. Accordingly, the video requests are completely handled by edge servers, streaming the content directly to the clients. No requests are forwarded to the central server, as it instead happens in the proxy-based approach whenever the proxy doesn't have a copy of the requested content. Even if more robust than the centralized solution, major disadvantages limit the diffusion of the proxy-based and CDN approaches. The former *translates* a single point of failure into many points of failure, fractioning central server load to more servers. The later may ensure high-quality services but it requires big investments for both network and servers deployment and management. Additionally, all these systems have scalability problems, that is, when the number of clients increases, the only way to satisfy all the incoming requests is to add new servers proportionally.

Hybrid architectures combine the employment of a centralized server with that of a peer-to-peer (P2P) network. Indeed, P2P technologies have been adopted for the deployment of important applications over the Internet, such as file sharing [71] and voice-over-IP (VoIP) [72]. Differently from file sharing, a P2P-VoD network must guarantee the video delivery to the end-user before rigid deadlines. In P2P-VoD, peers support the delivery of the video to other peers using a cache-and-relay strategy making use of their upload bandwidth so as to decrease server load and to avoid network congestions close to the server site. Advantages are a better use of resources and an increased system capacity that allow for the management of higher number of users. P2P networks are also used to realize video broadcast/multicast over the Internet [73]. This technology is attractive because the P2P paradigm has the intrinsic potential to scale with the number of active participants without requiring additional infrastructure deployments: a greater demand generates more resources.

In a peer-to-peer network each peer is free to join and leave the network without notice, bringing to the phenomena of peer churns. These peer dynamics are dangerous for VoD architectures, affecting the integrity and retainability of the service. In the past, many studies have addressed peer churns in file-sharing networks [74], [75], and some others focus on P2P-VoD systems proposing different techniques to avoid service disruption due to peer churns [76]-[78]. Differently from these works, we don't propose any new solution but analyses the user behaviour so as to develop models aimed at evaluating the impact of the peer churns on the system performance. Four models are then proposed. The first two rely on the Gilbert-Elliot model to represent the user connected and disconnected states; the third one is based on a fluidic analysis of the system; and the last one makes use of the queuing theory to represent how the video chunk download requests are processed by the system. The models are compared by computing the resource that system has to add on top of the P2P network to satisfy all the download requests. The importance of an accurate modelling of the churns lies on the possibility to analyse important relationships between system parameters, such as playback buffer length, peer request rate, peer average lifetime, and server upload rate, which can then be used to drive the dimensioning and optimization of system resources while assuring user satisfaction.

This chapter is organized as follows: section 4.2 illustrates a common peer-to-peer Video on Demand scenario, which represents the basis of our analysis. In Section 4.3, the proposed theoretical models are described and in Section 4.4 numerical analyses are presented. Section 4.5 draws final conclusions.

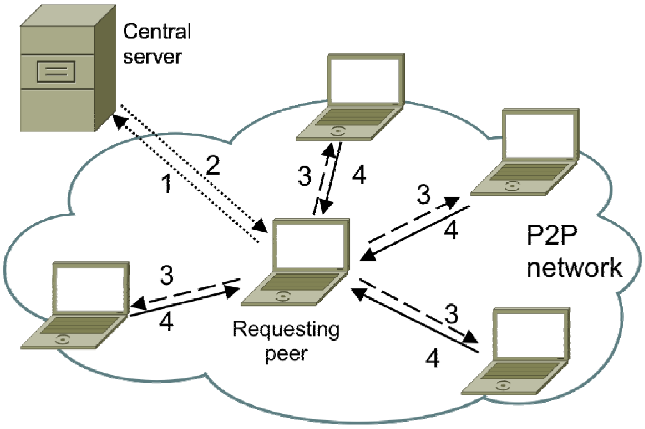

Figure 4.1. P2P-VoD system: 1 – Contact server, 2- Receive peer-list, 3- Create application level connections, 4- Send data

4.2 The P2P-VoD Scenario

In a typical P2P-VoD scenario a centralized server receives video requests whereas a number of peers download and upload the same content. This is referred to as a single-video approach and it differs from the multiple video approach because one peer can share only a video, which is the one it's playing back [79]. In case that all requested content cannot be provided by the peers, the server also streams the content accordingly.

Data and control information exchanges can be summarized in few steps. When a new peer joins the system, it contacts the server to know the available video contents. It chooses the video it is interested in and the server sends a list of possible peers that are viewing the chosen content; the peer then tries to create the necessary number of unicast connections with other peers to receive the content and start playing back. When a contacted peer had accepted a connection request, it starts to send useful data. This procedure is illustrated in Fig. 4.1.

Each peer has a playback buffer used for decouple network dynamics from video playing. If a contacted peer doesn't have the requested data at that moment or it doesn't reply to the contacting peer, the later starts creating another connection with the next peer according to the list provided by the server.

The server takes charge of distributing a refreshed peer list to all peers whenever necessary, assuming a central role in the coordination of the VoD service. The most critical problem in a P2P-VoD network is related to the dynamics of peer's participation. In a pure file-sharing network, it is not a serious problem: there are no deadlines to be respected and it may not be a vital matter if the file download takes more than the expected or desired time. Instead, in the scenario of streaming applications we are

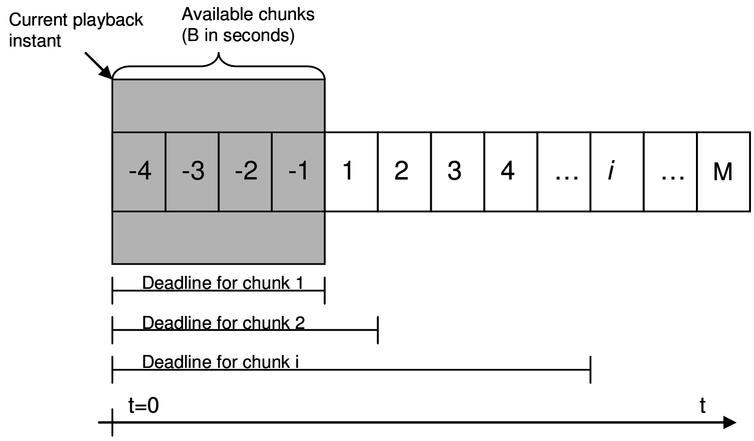

Figure 4.2: Playback buffer, video chunks and associated deadlines

considering, peer churns become an important issue which needs to be taken into account to make the system reliable enough to provide an acceptable QoS to the end-user.

The video content is divided in a sequence of video units, named chunks. To avoid playback interruptions, a peer must receive the correct sequence of these chunks before its playback deadline. Not to waste bandwidth, each peer can request only one chunk at time to one peer. We assume that each chunk is of the same transmission length T_{UT} (time to complete the transmission) and of the same playback length T_{UP} (time to finish the playback), both expressed in seconds. Typically T_{UT} is greater than T_{UP}, requiring more than one upstream peer (roughly T_{UT}/T_{UP} peers) per downloading peer on average to have a continuous playback of the video without server support. We assume that each peer access line has the same average upload bandwidth *U*, lower than the video streaming rate *R*. This is a frequent condition for Internet access in Small-Office Home-Office (SOHO) and domestic users, often characterized by asymmetric access lines.

Fig. 4.2 shows the streaming time-line. The peer has filled its playback buffer and is then starting the playing back. Chunks are enumerated in an increasing progressive way and the playback buffer *B*, measured in seconds, is of $4{*}T_{UP}$ in length in this example. Chunks from 1 to *M* are not available yet, and are in download phase from other peers at rate *U*. The deadline for every chunk *i* is:

$$D_i = B + (i-1)T_{UP} \tag{4.1}$$

Every time a disconnection occurs, the peer must contact a new available peer. We name the time necessary to complete a correct transmission Time-to-Redirect (TTR), as described in Fig. 4.3. On the contrary, the *unTTR* is the time wasted because of one disconnection and it is less or equal than TTR. For simplicity, we consider the worst case taking always *unTTR* equal to TTR. The TTR depends from many factors, such as: nearness of other peers, popularity of video content and network load.

Due to limited buffer capacity, peers can tolerate up to a maximum number of churns. When the total number of churns is becoming too high for a chunk transmission to a peer, the server takes part in the process by directly sending the chunk. In this scenario, it is interesting to evaluate which is the impact of churns to the whole system. In the following, we describe the proposed four theoretical models to represent peer churns in a P2P-VoD system.

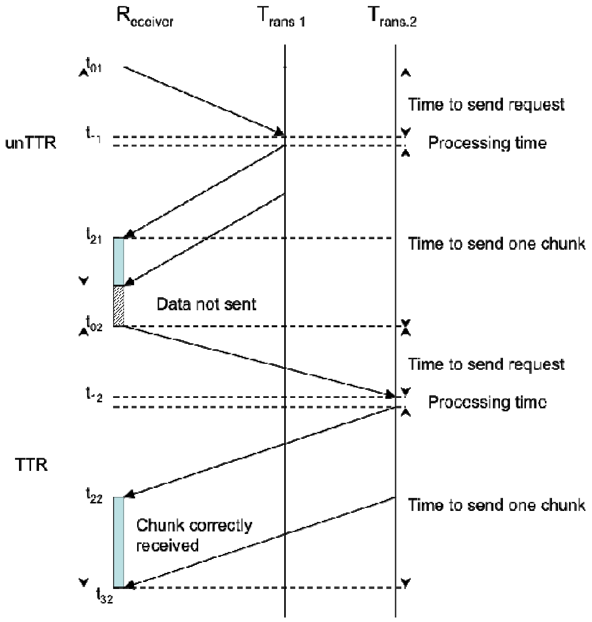

Figure 4.3: A peer churn event of length unTTR and successive

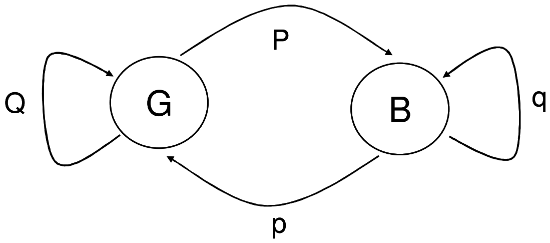

Figure 4.4: Gilbert-Elliot model

4.3 Models

In this section we present the proposed models. The first two are based on the Gilbert Elliot (GE) model; the third one relies on Fluidic analysis and the last is based on the Queueing theory.

- ***GE Model***

In this work, we initially model the peer behaviour using a two-state discrete-time process in which the time axis is measured in terms of *TTR* intervals. Such a process is then represented with a GE model [80],[81] drawn in Fig. 4.4. The transition probability *P* refers to the progress of the peer from the connected-state (good state G) to the disconnected-state (bad state B) during an interval *TTR*, whereas probability *p* refers to the inverse process.

Differently, *Q* and *q* refer to the probability to remain in the good and bad state, respectively, for an entire interval *TTR*. In our model, transition probabilities are changed time by time to represent changes in the user behaviour. This probability is taken randomly according to a uniform distribution because peer behaviour is considered stateless and peer participation is supposed very unpredictable. The uniform distribution is left constant for the entire session.

Based on the deadlines described in Section II, the maximum tolerable number of disconnections is defined as:

$$N_{DISC,i} = \frac{D_i}{TTR} \tag{4.2}$$

Each chunk has its own deadline, which has to be met not to interrupt video playback. The probability of to satisfy the deadline condition for a generic chunk i is:

$$\psi_i = \sum_{k=1}^{N_{DISC,i}} P^{k-1} * Q = P_{S,i} \tag{4.3}$$

This condition has to be fulfiled for every chunk that a peer is downloading. The probability to fulfil this condition is :

$$\psi = \prod_{i=1}^{M} \tag{4.4}$$

Considering the streaming rate R and the number of peers N into the system, the total bandwidth W_{TOTAL} requested by the whole system is:

$$W_{TOTAL} = R * N \tag{4.5}$$

Instead, the peers can provide an upload bandwidth *W_{PEER}* equal to:

$$W_{PEER} = \psi * N * U \tag{4.6}$$

Finally, the bandwidth that the peers are not able to guarantee is the difference between (4.5) and (4.6): this is the bandwidth W_{SERVER} requested to the server:

$$W_{SERVER} = W_{TOTAL} - W_{PEER} \tag{4.7}$$

- ***GE extended model***

The GE model is characterized by transition probabilities selected randomly according to a uniform distribution, which is kept constant during the entire video. However, recent studies [82] on user accesses over time, arrival rates and session lengths have shown that the user behaviour changes during the video playback session. It often happens that the user starts streaming the video and, after a while, he is not satisfied with the content then moves to another video. Accordingly, the probability that an user selects another video is a function of the time and it decreases as the total amount of played back video increases. Indeed, the probability of streaming interruption is very low after half of the video has been already seen. In particular, it has been proved that the cumulative distribution function of video session lengths is well-fitted by an exponential distribution.

Starting from these studies, we propose a GE model extension in which the probability of disconnection *P* is set according to an exponential distribution: in this way the stay-connect time of each peer is a monotonically increasing function of time, reflecting user trend to stay connected once a significant of video has already been watched. Probability of connection *p* is instead kept constant: its temporal variation's scale is very big if compared with disconnection probability variation, and for this reason it can be considered constant.

- ***Fluidic Model***

Recently, researchers have explored stochastic fluidic analytical models [83], [84] to model traffic in P2P networks. In these models, data transmission is seen like a fluid transferred through nodes, in a similar way to hydraulic models. Another study [78] develops a model for P2P-VoD in a broadcast environment. This model can be adapted to P2P-VoD with the hypothesis that peers in upload state can share all video in their memory, not only the first part. Peers can request aid from the server if the P2P network is not able to provide video data, which is the scenario we are considering in this paper.

The state diagram of a peer has 3 states: download, upload and depart, as shown in Fig. 4.5.

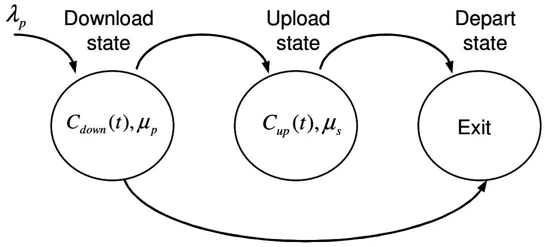

Figure 4.5: Peer state diagram

When a peer joins the system, it goes in download state and can receive the first part of the video by the P2P network. Therefore, if its playback buffer is full, it goes to the upload state where it can share video parts already downloaded. Finally, a peer can leave the system and moving into the depart state.

The final target is reducing server load in the download state using upload capabilities of peers. From queueing theory point-of-view, the all system can be approximated as a tandem queueing network with arrival and departure Poisson processes. Given:

λ_p	Arrival rate
$\mu_p = \dfrac{1}{First\_part\_length}$	Mean time in Download State
$\mu_s = \dfrac{1}{Second\_part\_length}$	Mean time in Upload state
γ_p	Mean Life Time
$C_{down}(t)$	Number of peers in Download state
$C_{up}(t)$	Number of peers in Upload state

it can be developed a simple fluid model to study the system evolution. Peers number in the first state can be calculated considering their exponential distribution, which is is proportional to ratio between peer's arrival rate and both mean life time and mean service time (4.8):

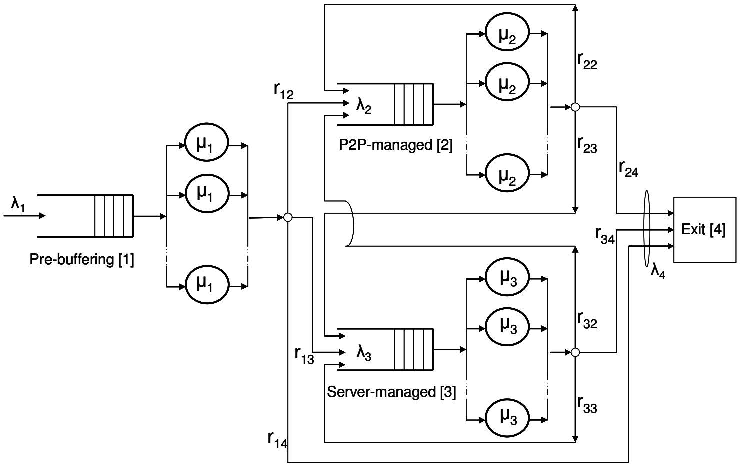

Figure 4.6: Proposed Queueing model of the P2P-VoD system

$$C_{down}(t) = \frac{\lambda_p}{\gamma_p + \mu_p}[1 - e^{-(\gamma_p + \mu_p)t}] \tag{4.8}$$

Instead, peers number variation in upload state is equal to the difference between peers coming from the download state and the peers going to the exit state:

$$\frac{d}{d(t)}\left[C_{up}(t)\right] = \mu_p C_{down}(t) - \min(\mu_s, \gamma_p) C_{up}(t) \tag{4.9}$$

The solution of differential equation (4.9) is the value of C_{up} as function of the time. The aggregate bandwidth of the P2P network W_{PEER} at time t is equal to $U*C_{up}(t)$ and bandwidth W_{SERVER} requested by central server is:

$$W_{SERVER} = [C_{down}(t) + C_{up}(t)] * R - C_{up}(t) * U$$

(4.10)

○ *Queueing model*

Queueing theory can be applied to a multiplicity of real problems, especially to transports and telecommunications fields, where each complex system is modeled by a set of queues connected each other. Each individual queue is called node and the state of a queueing network is defined by the simultaneous distribution of customers in each node. In open networks the input rate to a queue i is given by:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j r_{ji} \tag{4.11}$$

The term λ_{0i} is the arrival rate of tasks to i-th node from outside and r_{ji} are the routing probabilities that a served task is passed from node i to node j. The term λ_j is the arrival rate of tasks from internal nodes. A simple queueing network model can be constructed splitting the life cycle of peer in four different phases or states. The first state is a "pre-buffering state": peer joins the P2P network and buffers a certain quantity of data before to start video playing. When its buffer is full it can be routed to the "P2P-managed state", to "Server-managed state" or can leave the system going in "Exit state". Each state is represented by an M/M/∞ queue except for the exit state. The proposed queuing model is shown in Figure 4.6.

The M/M/∞ queue model is chosen for its analytical tractability. The first queue exactly models the startup delay necessary to fill up the playback buffer. The aim is to collect enough data before starting the video playback to decouple the playback time from the transmission time.

video playing to decouple playing from network dynamics. The buffer length is fixed, so that the service rate is constant:

$$\mu_1 = \frac{1}{B} \tag{4.12}$$

When a peer has filled its buffer, it leaves the first queue and can be routed toward others queues or leave the system. Routing probability depends on the probability to leave the system α and probability to receive data from others peers P_{hit}. For each state, (4.13) has to be fulfilled as well as the constraint about outgoing routing probabilities for every *i*:

$$\sum_j r_{ij} = 1 \tag{4.13}$$

Additionally, the following routing probabilities apply:

$$r_{12} = P_{hit} \tag{4.14}$$

$$r_{13} = 1 - \alpha - P_{hit} \tag{4.15}$$

$$r_{14} = \alpha \tag{4.16}$$

Exit state could be considered as another queue with service rate unitary: in truth, it's important to calculate only the overall arrival rate to evaluate model dynamics.

$$\begin{cases} r_{12} = r_{22} = r_{32} \\ r_{13} = r_{23} = r_{33} \\ r_{14} = r_{24} = r_{34} \end{cases} \tag{4.17}$$

The mean total number of peers in the system is:

$$\overline{X}_{TOT} = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} + \frac{\lambda_3}{\mu_3} = \rho_1 + \rho_2 + \rho_3 \tag{4.18}$$

Hit probability is calculated dynamically and is proportional to the arrival rate in queue 3 and in exit state.

$$P_{hit} = 1 - \frac{\lambda_3}{\overline{X}_{TOT}} - \frac{\lambda_4}{\overline{X}_{TOT}} \tag{4.19}$$

Considering the number of peers ρ_i in each queue i, the bandwidth requested by central server is:

$$W_{SERVER} = (\rho_1 + \rho_2 + \rho_3) * R - \rho_2 * U \tag{4.20}$$

Finally, we need to specify the sense of mean service time in queue 2 and 3: every time-step long as mean service time, the next peers' status is set in relationship to the number of peers in queue 2 and 3. If the P2P system contains a sufficient number of peers so that the hit probability is high, this situation influences probability of routing toward P2P-managed state. Otherwise, P_{hit} decreases and it's more probable that a peer will forward to Server-managed state. Notice also that is not possible to have peers in waiting line because there is always a servant free in a M/M/∞ queue.

4.4 Simulations

We have performed extensive simulations with different scenarios. The objective of the simulation analysis is to investigate the models behaviour varying the system parameters in order to assess the usefulness of such models in supporting the design and configuration of P2P-VoD architectures. Herein, we present the results when applying the following streaming parameters: transmission of video sequences of 100-minute length at 800Kbpsand an upload rate *U* of 600Kbps. We choose these values according to the condition *U*<*R*, that reflects the most common situation of Internet access lines as explained in section 4.2.

The simulations with the GE model has been conducted with a total of 50 peers in the system and changing the stay-connect probability *Q* every 10 seconds. This probability has been chosen according to a uniform distribution with different ranges, as shown in Table I. The connection probability *p* has been kept constant and equal to 0.5 during all simulations. In the GE extended model, the disconnection probability *P* follows the exponential distribution (4.21)

$$P(t) = T * e^{-\left(\frac{t}{T}\right)} \tag{4.21}$$

with the parameter *T* set so that the complementary stay-connect probability *Q* has the mean values of Table V.

Table V: Ranges of the uniform distribution for the probability *Q* of the GE model.

Range	Values	Mean
#1	0÷0.1	0.05
#2	0÷0.2	0.1
#3	0÷0.3	0.15
#4	0÷0.4	0.2
#5	0÷0.5	0.25
#6	0÷0.6	0.3
#7	0÷0.7	0.35
#8	0÷0.8	0.4
#9	0÷0.9	0.45

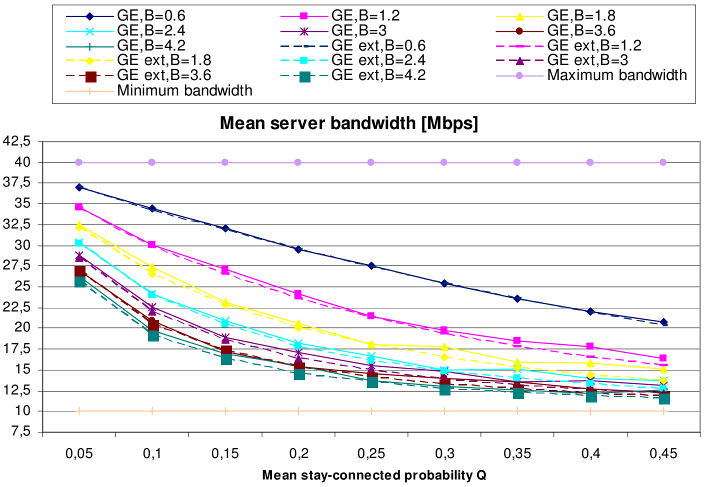

Figure 4.7: GE models comparison for different values of buffer length

To evaluate the effectiveness of these two models, we have computed the requested server download rate at varying disconnection probabilities *P*. Fig. 4.7 shows the results for the two models when the time-to-repair TTR has been set to 300msec and changing the buffer length from 0.6 to 4.2 sec.

It can be noted that the two models show similar behaviours, as it was expected since the models are basically the same except the distribution of the connection probability. The shape of the plots shows that increasing the buffer length brings to lower requested bandwidth values. This is due to the fact that the deadline for each chunk is less stringent, allowing for finding an active peer from which successfully download the chunk. The curves are convex so that a higher benefit is obtained by increasing the buffer length at low values of the *Q* probability. The figure also shows that the total amount of average server bandwidth converges towards 10Mbps ("minimum server bandwidth" in the figure), which indeed is the difference between the W_{TOTAL} bandwidth of 40Mbps and the maximum theoretical bandwidth provided by the peers W_{PEER}, which is of 30Mbps. Overall, this figure is an handy tool that helps the designer in finding the server resources that are required to satisfy the user requests on the basis of the playback buffer length and as far as the operator is able to estimate the peer stay-connected probability. Note that the curve "maximum server bandwidth" in the figure represents the amount of server bandwidth that would be necessary without the support of the peer-to-peer network.

In the Fluidic and the Queuing Network models, one of the key parameters is the mean time a peer spends in the system, which is the peer Mean Time in the System (MTS). Whereas for the first model it is directly set by selecting the value of Mean Life Time γ_p, for the Queueing Network model the MTS is indirectly set through the probability to leave the system α, the sampling step Δ and number of simulation samples N_s according to the following formula:

$$MTS = \sum_{i=0}^{N_S} [(1-\alpha_i)^i \alpha_i \cdot i \cdot \Delta] \tag{4.22}$$

(4.22) has been used to find parameters values to achieve the desired MTS. As to the parameter P_{hit}, it has been initialized to 0.9, whereas successive values are dynamically calculated according to the model evolutions. For the analysis of these models we have computed the mean server download bandwidth requested by each peer, while varying the following parameters: MTS and input arrival rate λ (L in Fig. 4.8). These two parameter affect the number of peers into the system, which then cannot be directly set by us as in the GE models.

Fig. 4.8 shows the requested bandwidth for the Fluidic model. Note that this time the resulting value has been divided by the number of peers in the system, which is different for any combinations of system parameters (see Table VI). In this figure we are also showing the upper and lower bandwidth limits: 800Kbps is the rate requested to the server when no one peer is able to share video data, whereas 200 Kbps is the difference between the video rate R and the maximum upload rate U, which corresponds to the amount of bandwidth that should be provided by the server when all the active peers are successfully sending video content to another peer.

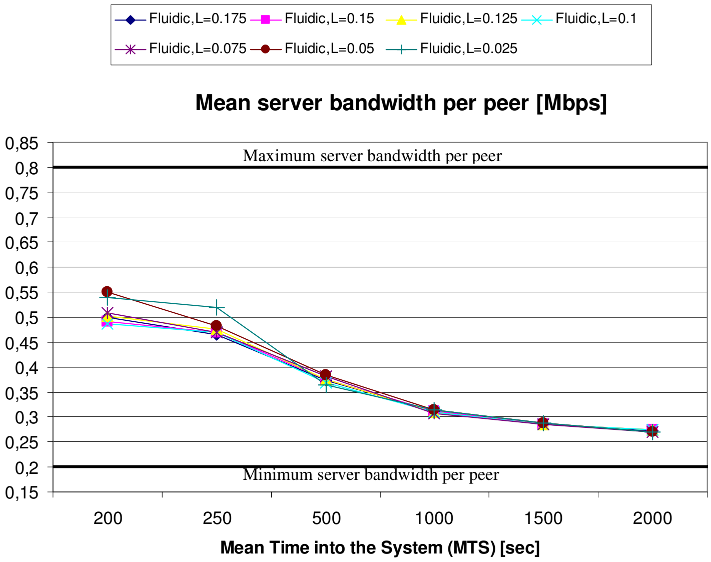

Figure 4.8: Mean server bandwidth requested by each peer in the Fluidic model with different peer input rates

The shape of the plots shows a decreasing bandwidth requested as a function of MTS, and implicitly with the increasing number of peers into the system: this behaviour confirms the implicit feature of system scalability of P2P systems. In fact, a bigger number of peers into the system generates more resources (upload bandwidth), reducing the bandwidth requested to server per peer into the system.

Table VI: Mean number of peers measured in the Fluidic Model .

Input rate	Mean number of peers					
0.175	34	41	80	146	198	240
0.150	29	36	69	125	170	205
0.125	24	29	58	104	141	171
0.100	19	23	46	83	113	137
0.075	14	18	34	62	85	102
0.050	9	11	23	42	56	68
0.025	4	6	11	21	28	34
MTS	200	250	500	1000	1500	2000

4.5 Conclusions

In this chapter we have presented three mathematical models for the evaluation of the peer churn impact on the server resources in P2P-VoD systems. In the first model, the behaviour of each peer is represented by means of the Gilbert-Elliot model, where the two states are associated to the connected and disconnected states. The second and third models use a very different approach with respect to the GE one: a constant number of peers joins the system and the resources requests are related to the effective number of peers inside the system.

The simulations have shown that these models are an effective tool that help the designer in finding the server resources that are required to satisfy the user requests on the basis of the playback buffer length, as far as the operator is able to estimate the peer stay-connected probability. The longer the time each peer spend in the system, the lower the resource required to the server. In fact, an increase in the average stay-connected interval decreases the probability to waste time sending only useless partial chunks from peer to peer, which need to be resent from the beginning by another peer (if available) or by the server.

Chapter 5. Conclusions

We have started this thesis presenting the principles, the standardization efforts and the actual scenario of next generation networks. The rapid evolution of the user needs and market trends force flexibility and dynamicity requirements in the provisioning of new services. NGN are expected to satisfy these requirements, that are intended as the capability of the network to bind on-demand services and related resources at user request and according to his profile. In our opinion and to the best of our knowledge, the service oriented approach can be the most effective solution to implement NGN architectures which follows flexibility and dynamicity requisites. Until now, research and development interests have focused on service stratum, whereas the transport stratum has not received adequate investigation actions. We think the transport stratum could have an important role to satisfy NGN expectations.

We have proposed an architecture of four different layers for the implementation of the transport stratum in NGN, following a SOA-oriented design. The SOA approach can lead up to the development of a more and dynamic transport stratum, without any impact on the upper service stratum. In our work, we have specifically referred to the DS-TE architecture, isolating the elementary services that can be composed to build complex services, which are then subsequently exposed to the service stratum. The access to real devices is guaranteed by a Web Service Proxy that is able to directly communicate with physical devices thanks to an object-oriented library.

We have conducted some evaluations, by implementing a fully working prototype of the proposed system using the Web Services technology and making relevant experiments. During the system development and deployment, we have experienced the following advantages that characterize the proposed solution:

- the design of a complex service starting on the single elementary services is a simple and intuitive practice from both a technician that knows very well the transport technologies and a high-level language developer;
- the time needed to develop and deploy a complex service are reduced with respect to the whole development of ad-hoc services;
- the adoption of standard and easy-understandable interfaces contributes to improve the learnability of the framework;

- given a complex service in execution, an elementary service can be added in a dynamic way, without affecting the complex service execution flow.

The major contribution of this paper is a SOA-oriented architecture for the implementation of the transport stratum, which resulted to provide the following major advantages to the network operator:

- the use of a common formalism for the definition of low level telecommunication services;
- the use of common interfaces that don’t require the operators to know the inner details of each single service;
- telco services at the higher levels can be obtained as a composition of other services in the lower layers using a composition and coordination logic.

By applying the SOA design, we have obtained a list of elementary services which are expected to be implemented at the transport stratum. However, these are not mandatory for the adoption of the proposed architecture and the exact list may change from an operator to another depending on the internal policies. Still, we believe that a core set of services and relevant interfaces need to be discussed and agreed on in a consensus meeting (within standardization activities) to improve the interoperability among intra-domain network components and among separate network domains, as well as to accelerate the deployment. The results of this work represent a first contribution in this direction.

The second part of this thesis illustrates the problem of bandwidth management in NGN architectures, with specific regard to DS-TE scenario where the setting of the bandwidth constraints is an important parameter of Traffic Engineering approach. DS-TE architectures implement TE in a per-class basis making use of bandwidth constraint models, which control the amount of resources assignable to each traffic class per link and influence the execution of the routing algorithms, then representing a powerful tool to drive the utilization of the resources in the network and the fulfilment of the QoS requirements.

Starting form this assumption, a new bandwidth management solution is proposed, which relies on a distributed and self management approach. A self-management module is implemented in each node of the network: it monitors the unreserved bandwidth in adjacent nodes and adjust the local bandwidth constraints so as to reduce the differences in the unreserved bandwidth of neighbour nodes. Accordingly, we smooth abrupt differences in bandwidth availability along possible paths, which are frequently due to static settings of bandwidth constraints. Due to the distributed nature of the proposed solution, the adjustments can be frequently introduced, allowing for a quick adaption of the network to the traffic changes. The proposed solution has been compared with static bandwidth constraint settings in terms of resulting bandwidth blocking rates. The experimental results showed that it allows for obtaining an increase in the number of LSPs that the network is able to accommodate. When adopting this strategy, the network operator has just to set a few weights and leave the network autonomously correct the setting as the traffic and link occupation levels change.

Finally, we have addressed the problem of the video transmission, specifically the Video On Demand service delivery. Realizing a VoD system using the Internet requires architectures tailored to video characteristics. Peer-to-peer networks architectures (P2P) are becoming more and more popular in video content delivery services, such as TV broadcast and Video on Demand (VoD), thanks to their scalability feature. Such characteristic allows for higher numbers of simultaneous users at a given server load and bandwidth with respect to alternative solutions. However, great efforts are still required to study and design reliable and QoS-guaranteed solutions.

Within the scenario of P2P-based VoD services, we propose four models of the peer behaviour to evaluate peer churn, which impact on the system performances. They are based on: the Gilbert-Elliot chain, the fluidic representation of the user behavior and a queuing analysis of the system. The models are compared by computing the resources the system has to add on top of the P2P network to satisfy all the download requests; they provide an important tool for service providers to evaluate resources to add on the system architecture with the aim to guarantee an adequate level of quality of service for final users.

Acknoledgements

The research activity carried on this thesis has been supported within the IKNOS project funded by the Italian Ministry of Education and Research and developed by the University of Cagliari, CNIT and Tiscali.

Special thanks go to Dr. Luigi Atzori, for the constant support during all my activities in these three years as PhD student in the CNIT Multimedia Communications Laboratory at the University of Cagliari.

I also would like to thank all the staff of MCLab and all the people that supported me both from a scientific and human point of view along this period of my life.

References

[1] ITU-T Rec. Y.2001: General overview of NGN. [Online]. Available: http://www.itu.int/ITU-T/studygroups/com13/ngn2004/

[2] Telecommunication and Internet converged Services and Protocols for Advanced Networking (TISPAN) standardization body. [Online]. Available: http://www.etsi.org/tispan/

[3] European Telecommunication Standard Institute (ETSI) - ES 282 001 Specification. [Online]. Available: http://electronics.ihs.com/collections/etsi/

[4] The TINA Consortium. URL: http://www.tinac.com/

[5] R.Branden, D.Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview.", IETF RFC 1633, June 1994

[6] S.Blake et al., "An Architecture for Differentiated Services.", IETF RFC 2475, Dec. 1998

[7] E.Rosen et al., "MultiProtocol Label Switching Architecture", IETF RFC 3031, 2001

[8] F. Le Faucheur, "Protocol Extensions for Support of DiffServ-aware MPLS Traffic Engineering", RFC 4124, June 2005

[9] Organization for the Advancement of Structured Information Standards (OASIS). [Online]. Available: http://www.oasis-open.org/home/index. php

[10] T. Erl, *Service-Oriented Architecture: Concepts, Technology and Design*. Upper Saddle River, NJ,USA: Prentice Hall PTR, August 2005.

[11] D. Griffin and D. Pesch, “A Survey on Web Services in Telecommunications,” *IEEE Communications Magazine*, vol. 45, no. 7, pp. 28–35, July 2007.

[12] E. Mikoczy, S. Schumann, H. Stokking, M. van Deventer, and O. Niamut, “Combinational services of NGN based IPTV,” in *Proc. of 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, TridentCom 2009*, April 2009, pp. 1 –8.

[13] ITU-T Rec. M.3060, “Principles for the Management of Next Generation Networks.” [Online]. Available: http://www.itu.int/rec/ T- REC- M.3060- 200603- I

[14] The Moriana Group, “SDP 2.0 - Service Delivery Platform in the Web 2.0 Era,” September 2008. [Online]. Available: http: //www.morianagroup.com/

[15] “Perspectives 2010: The Official TM Forum Yearbook.” [Online]. Available: http://www.tmforum.org/Perspectives2010The/8157/home.html

[16] H. Ohnishi, Y. Yamato, M. Kaneko, T. Moriya, M. Hirano, and H. Sunaga, “Service Delivery Platform for Telecom-Enterprise-Internet Combined Services,” in *Proc. of IEEE GLOBECOM Conference*, 2007, pp. 108–112.

[17] A.-J. Moerdijk and L. Klostermann, “Opening the Networks with Parlay/OSA: Standards and Aspects behind the APIs,” *IEEE Network*, vol. 17, no. 3, pp. 58–64, May-June 2003.

[18] Parlay-X 3.0 Specifications. [Online]. Available: http://portal.etsi.org/ docbox/TISPAN/Open/OSA/

[19] ETSI ES 201 915-1 Specification: Open Service Access, Application Programming Interface. [Online]. Available: http://www.etsi.org/deliver/etsi es/201900 201999/20191501/01. 01.01 50/es 20191501v010101m.pdf

[20] T. Moriya and J. Akahani, “Application Programming Gap between Telecommunication and Internet,” *IEEE Communications Magazine*, vol. 48, no. 8, pp. 96 –102, Aug. 2010.

[21] C. E. A. Mulligan, “Open API standardization for the NGN platform,” *IEEE Communication Magazine*, vol. 47, no. 5, pp. 108–113, 2009.

[22] C. Mulligan, “Embracing Open Source Methods for the Standardization of NGN Services and Enablers,” in *Proc. of Second International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST ’08.*, Sept. 2008, pp. 628 –633.

[23] S. W. Choi and S. D. Kim, “A Quality Model for Evaluating Reusability of Services in SOA,” in *Proc. of 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, July 2008, pp. 293 –298.

[24] V. Narayanan, “Building Reusable Services,” *The SOA Magazine*, vol. XXVIII, April 2009.

[25] R. Ahmed, N. Limam, J. Xiao, Y. Iraqi, and R. Boutaba, “Resource and Service Discovery in Large-scale Multi-domain Networks,” *IEEE Communications Surveys & Tutorials*, vol. 9, no. 4, pp. 2 –30, 2007.

[26] N. Limam and R. Boutaba, “QoS and Reputation-Aware Service Selection,” in *Proc. of IEEE Network Operations and Management Symposium, NOMS 2008*, 7-11 2008, pp. 403–410.

[27] W. Huang and X. Li, “A SOA Based Framework of Service Ability Description and Selection Methods for Application Integration in Manufacturing Domain,” in *Proc. of International Conference on Interoperability for Enterprise Software and Applications China, IESA ’09.*, 2009, pp. 206 –211.

[28] S. Li, H. Chen, and X. Chen, “A Mechanism for Web Service Selection and Recommendation Based on Multi-QoS Constraints,” in *Proc. of 6th IEEE World Congress on Services, 2010*, July 2010, pp. 221 –228.

[29] N. Blum, S. Dutkowski, and T. Magedanz, “InSeRt.” Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 21–30.

[30] C. Rolland, R. Kaabi, and N. Kraiem, “On ISOA: Intentional Services Oriented Architecture,” in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, J. Krogstie, A. Opdahl, and G. Sindre, Eds. Springer Berlin / Heidelberg, 2007, vol. 4495, pp. 158–172.

[31] T. Margaria and B. Steffen, “Agile IT: Thinking in User-Centric Models,” in *ISoLA*, 2008, pp. 490–502. [Online]. Available: http: //dx.doi.org/10.1007/978-3-540-88479-8 35

[32] B. Steffen, T. Margaria, R. Nagel, S. Jo ̈rges, and C. Kubczak, “Model- driven development with the jABC,” in *Proc. of the 2nd international Haifa verification conference on Hardware and software, verification and testing, HVC’06*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 92–108.

[33] N. Blum, T. Magedanz, J. Kleessen, and T. Margaria, “Enabling eXtreme Model Driven Design of Parlay X-based Communications Services for End-to-End Multiplatform Service Orchestrations,” in *Proc. of 14th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS ’09*. IEEE Computer Society, 2009, pp. 240–247.

[34] “Oracle Communications Services Gatekeeper.” [Online]. Available: http://www.oracle.com/us/industries/communications/046870.html

[35] A. Sur, F. Schaffa, J. McIntyre, J. Nogima, M. Alexander, and P. Dettori, “Extending the Service Bus for Successful and Sustainable IPTV Services,” *IEEE Communications Magazine*, vol. 46, no. 8, pp. 96 –103, 2008.

[36] J. Fiedler, C. Riede, and T. Magedanz, “Converged Media and Communication Services,” in *Proc. of First Conference on IT Revolutions, 2008*, 17-19 2008, pp. 1 –7.

[37] “3GPP IMS Specifications .” [Online]. Available: http://www.3gpp.org/ article/ims

[38] X. Zhang, J. Song, and H. Qu, “Research on the Architecture of Telecom Services Platform Based on IMS and SOA,” *Proc. of Joint Conferences on Pervasive Computing, JCPC 2009*, pp. 523 –528, Dec. 2009.

[39] H. Khlifi and J.-C. Gregoire, “IMS for Enterprises,” *IEEE Communica- tions Magazine*, vol. 45, no. 7, pp. 68 –75, July 2007.

[40] H. Lu, Y. Zheng, and Y. Sun, “The Next Generation SDP Architecture: Based on SOA and Integrated with IMS,” in *Proc. of the*

Second International Symposium on Intelligent Information Technology Application, IITA '08, Washington, DC, USA, 2008, pp. 141–145.

[41] "Service Oriented Legacy Architecture (SOLA)." [Online]. Available: http://www.soa.com/products/sola/sola mainframe soa/

[42] "IBM IMS SOA Integration Suite." [Online]. Available: http: //www-01.ibm.com/software/data/ims/soa-integration-suite/

[43] B. Thurm, "Web Services for Network Management - A Universal Architecture and Its Application to MPLS Network," in *Proc. of 27th Annual IEEE International Conference on Local Computer Networks, LCN' 02*, 2002, pp. 463–472.

[44] O. Younis and S. Fahmy, "Constraint-based Routing in the Internet: Basic Principles and Recent Research," *IEEE Communications Surveys & Tutorials*, vol. 5, no. 1, pp. 2 –13, quarter 2003.

[45] E. Rosen, A. Viswanathan, and R. Callon, "RFC3031: Multiprotocol Label Switching Architecture," United States, 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3031.txt

[46] R. Neisse, R. L. Vianna, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "Implementation and Bandwidth Consumption Evaluation of SNMP to Web Sevices Gateways," in *Proc. of IEEE/IFIP Network Operations and Management Symposium, NOMS*, 2004, pp. 715–728.

[47] R. Vianna, M. Almeida, L. Tarouco, and L. Granville, "Evaluating the Performance of Web Services Composition for Network Management," in *Proc. of IEEE International Conference on Communications, ICC ' 07*, 2007, pp.1943–1948.

[48] R. Enns, "NETCONF Configuration Protocol," RFC 4741 (Proposed Standard), Internet Engineering Task Force, Dec. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4741.txt

[49] T. Goddard, "Using NETCONF over the Simple Object Access Protocol (SOAP)," RFC 4743 (Proposed Standard), Internet Engineering Task Force, Dec. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4743.txt

[50] M. Wasserman and T. Goddard, "Using the NETCONF Configuration Protocol over Secure Shell (SSH)," RFC 4742 (Proposed Standard), Internet Engineering Task Force, Dec. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4742.txt

[51] P. Anedda and L. Atzori, "Network Administration Using Web Services," in *Proc. of IEEE Global Telecommunications Conference, GLOBECOM 2009*, Dec. 2009, pp. 1 –6.

[52] H. Hanrahan, *Network Convergence: Services, Applications, Transport, and Operations Support*. New York, NY, USA: John Wiley & Sons, Inc., 2007.

[53] "TeleManagement Forum (TMF), New Generation Operations Systems and Software(NGOSS) Solution Framework." [Online]. Available: http://www.tmforum.org/SolutionFrameworks/8428/home.html

[54] K. Kotsopoulos, P. Lei, and Y. F. Hu, "Managing NGNs using the SOA Philosophy," in *Proc. of First ITU-T Kaleidoscope Academic Conference Innovations in NGN: Future Network and Services, K-INGN 2008*, 12-13 2008, pp. 47–54.

[55] ——, "A SOA-based Information Management Model for Next-Generation Network," in *Proc. of International Conference on Computer and Communication Engineering, ICCCE 2008*, 13-15 2008, pp. 1057 –1062.

[56] "Apache ServiceMix Enterprise Service Bus." [Online]. Available: http://servicemix.apache.org/home.html

[57] J. L. Salina and P. Salina, *Next Generation Networks - Perspectives and Potentials*. New York, NY, USA: John Wiley and Sons, Ltd, 2007.

[58] Faucheur, F.L., Lai, W.: Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering. RFC 3564 (2003)

[59] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., Xiao, X.: Overview and Principles of Internet Traffic Engineering. RFC 3272 (2002)

[60] Miyazawa, M., Ogaki, K., Otani, T.: Multi-layer Network Management System with Dynamic Control of MPLS/GMPLS LSPs based on IP Flows. In: Network Operations and Management Symposium, NOMS 2008, IEEE, pp. 263–270 (2008)

[61] Manousakis, K., Baras, J. S., McAuley, A., Morera, R. (2005). Network and Domain Autoconfiguration: A Unified Approach

for Large Dynamic Networks, IEEE Communication Magazine, August 2005

[62] Boutaba, R., Martin-Flatin, J.P., Hellerstein, J.L., Katz, R.H., Pavlou, G. (2010). Guest Editorial Recent Advances in Autonomic Communications, IEEE J. On Selected Area in Comm., Vol. 28, No. 1, January 2010

[63] Lee, H., Choi, G., Kim, H., Lee, K. (2008). A nature-inspired QoS routing algorithm for next generation networks, Fourth International Conference on Autonomic and Autonomous Systems, 2008

[64] Tizghadam, A., Leon-Garcia, A. (2010). Autonomic Traffic Engineering for Network, IEEE J. On Selected Area in Comm., Vol. 28, No. 1, January 2010

[65] Zheng, W. (2001). Internet QoS: Architectures and Mechanisms for Quality of Service, Morgan Kaufmann Publishers, March, 2001

[66] Onali, T., Atzori, L. (2008). Bandwidth Management in DS-TE Networks: A Genetic Algorithms based Approach, IEEE Globecom Conference, New Orleans, USA, November 2008

[67] Onali, T., Atzori, L. (2008). Traffic Classification and Bandwidth Management in DiffServ-aware Traffic Engineering Architectures, IEEE - ICC, Bejiing, China, May 19-23, 2008

[68] Cisco White Paper (2008,June,16). "Approaching the Zettabyte Era". Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481374.html

[69] H.Schwark, D.Marve, and T.Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol.17, no. 9, pp 1103-1120, Sep. 2007.

[70] F.Thouin, M.Coates, “Video-on-demand Networks: Design Approaches and Future Challenges”, IEEE Network, Vol.21,no. 2 pp 42-48, March-Apr. 2007.

[71] BitTorrent, www.bittorrent.com

[72] Skype, www.skype.com

[73] J.Liu, S.G.Rao, Bo Li,H.Zhang, “Opportunities and Challenges of Peer to Peer Internet Video Broadcast”, Proc. of IEEE, vol. 96, pp 11-24, Jan. 2008

[74] J.Tian,Y.Dai, “Understanding the Dynamic of Peer-to-Peer Systems”, International Workshop on Peer-To-Peer Systems (IPTPS 2007). Available: http://www.iptps.org/papers-2007/TianDai.pdf

[75] D.Stutzbach, R.Rejaje, "Understanding Churn in Peer-to-Peer Networks", Proc. of the 6th ACM SIGCOMM conference on Internet Measurement, pp 189-202, Sep.2006.

[76] Y.Guo, S.Mathur, K.Ramaswamy, Y.Shengchao, B.Patel: “PONDER: Performance Aware P2P Video-on-demand Service”, GLOBECOM ’07 IEEE, pp 225-230, Washington DC, Nov 07

[77] Z.Liu, Y.Shen, S.Panwar, K.W.Ross, Y.Wang: “Efficient substream encoding and transmission for P2P Video on Demand”, Packet Video 2007, pp 143-152, Lausanne, Switzerland, Nov 07

[78] K.M.Ho, K.T.Lo: "A Simple Model for Peer-to-Peer Video-on-Demand System in Broadcast Environment", International Conference on Information Networking, ICOIN 2008, pp 1-5, January 2008

[79] C.Huang, J.Li, K.W.Ross: "Peer-Assisted VoD: Making Internet Video Distribution Cheap", International Workshop on Peer-to-Peer systems (IPTPS), Redmond WA,2007

[80] E.N. Gilbert: "Capacity of a burst-noise channel", Bell Systems Technical Journal,39:1253–1265, September 1960

[81] E. O. Elliot: "Estimates of error rates for codes on burst-noise channels", Bell Systems Technical Journal, 42:1977–1997, September 1963

[82] Yu, D.Zheng, Zhao,W.Zheng: "Understanding User Behavior in Large-scale Video-on-Demand Systems", Proceedings of EuroSys2006, pp 333-344, Leuven, Belgium, April 18-21, 2006

[83] R.Kumar, Y.Liu, K.Ross :"Stochastic Fluid Theory for P2P Streaming Systems", IEEE International Conference on Computer

Communications, INFOCOM 2007, pp 919-927, May 2007

[84] Y.Lu, An Zhang, H.He, Z.Deng: "Stochastic Fluid Model for P2P Content Distribution Networks", Proceedings of Autonomous Decentralized Systems, ISADS 2005, pp 707-712, April 2005