Centro de Informática U·F·P·E

Pós-Graduação em Ciência da Computação

"An Architecture for Providing End-to-End QoS-based Advanced Services in the Internet"

Por

Carlos Alberto Kamienski

Tese de Doutorado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, FEVEREIRO/2003

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS ALBERTO KAMIENSKI

"AN ARCHITECTURE FOR PROVIDING END-TO-END QOS-BASED ADVANCED SERVICES IN THE INTERNET"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR:

DJAMEL FAWZI HADJ SADOK, Ph.D.

RECIFE, FEVEREIRO/2003

Universidade Federal de Pernambuco - UFPE
Centro de Informática - CIn
Doutorado em Ciência da Computação
Redes de Computadores

An Architecture for Providing End-to-End QoS-based Advanced Services in the Internet

By:
Carlos Alberto Kamienski

Thesis submitted to the Computer Science Center of the Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor: Prof. Djamel Sadok

Recife, February 2003.

Abstract

There is currently a strong trend for evolving the Internet into a converged network, able to deal with advanced voice and video applications in addition to the more traditional data applications. A prerequisite for this evolution is transcending the limitations of the best effort service, which gives the same forwarding treatment to the packets of all users, by offering services with Quality of Service (QoS) guarantees for the new and more stringent applications. In order to be effective, QoS guarantees need to be constant along the entire path between source and destination, i.e. end-to-end, across several domains often with different administrations and technical characteristics. An important point stressed by this work is that deploying advanced services is not a matter that can be resolved only by implementing QoS technologies that try to resolve the problem at the router level.

This thesis addresses the issue of providing QoS in the Internet. A three-plane architecture, Chameleon, is proposed for offering QoS-based advanced end-to-end services in the Internet. The service plane creates an abstract model of the network, so that every domain has a similar external interface. Its main functions are service definition, service negotiation and resource management. In the operation plane, domains implement negotiated services through some specific QoS technology. The monitoring plane is orthogonal to the other planes and its main responsibility is assuring that services are operating within the agreed quality levels.

Service definition in Chameleon refers to the activity of precisely explaining the semantics of a service· This definition allows every domain to understand service requirements in the same way, so that a unique end-to-end behaviour may be achieved, no matter how many different domains are cooperating to deploy it or the sort of different QoS technologies domains are using to implement it. Service negotiation refers to the process whereby domains communicate to each other in order to deploy services in the Internet. It involves verifying the feasibility of deploying a given service through some path (sequence of domains) in terms of its required performance guarantees. Advanced services may generate frequent changes on traffic volumes and routes, for meeting services' requirements. Therefore, *dynamic* service negotiation is necessary, based upon efficient, scalable and fair negotiation models. Chameleon allows the utilization of different negotiation models, such as the cascade, hub and hierarchical models.

A comparative evaluation of the cascade, hub and hierarchical negotiation models was carried out and the results yielded enlightening conclusions. The analyzed scenarios showed that the hierarchical model is the best alternative for a large number of domains, when comparing efficiency, fairness and scalability. It is also able to provide the right financial incentives for deploying QoS and resolve some of the current problems of the interconnection of domains in the Internet. When the number of domains is small, the cascade model proves to be superior, because of its simplicity. The hub model is similar to the cascade one in terms of benefits provided to the negotiation, but it raises too many scalability concerns.

Resumo

Atualmente existe uma tendência de a Internet evoluir para uma rede convergente, que pode tratar aplicações avançadas de voz e vídeo juntamente com as aplicações mais tradicionais de dados. Um pré-requisito para essa evolução é transcender os limites impostos pelo serviço de melhor esforço, que trata igualmente os pacotes de todos os usuários, através da implantação de serviços com garantias de Qualidade de Serviço (QoS) para as aplicações mais exigentes. Essas garantias devem estar presentes em todo o trajeto do tráfego entre fonte e destino, ou seja, fim-a-fim, passando por vários domínios (redes) com administrações e características técnicas diversas. A tônica deste trabalho é que implantar serviços avançados é uma questão que não pode ser resolvida somente pela implementação de tecnologias de QoS que tentam solucionar o problema no nível dos roteadores.

Esta tese trata da questão de oferecer QoS na Internet. A arquitetura Chameleon é proposta para auxiliar a implantação de serviços avançados fim a fim baseados em garantias de QoS na Internet. Ela é dividida em três planos lógicos. O plano de serviços define um modelo abstrato da rede para que a interface externa de todos os domínios seja similar. As suas principais funções são a definição de serviços, negociação de serviços e gerenciamento de recursos. No plano de operação, os domínios implementam os serviços que foram negociados através de alguma tecnologia de QoS. O plano de monitoramento é ortogonal aos outros planos, sendo responsável por assegurar que os serviços estão operando dentro dos níveis de qualidade que foram negociados.

A definição de serviços em Chameleon se refere à atividade de esclarecer precisamente a semântica de um serviço. Essa definição permite que cada domínio tenha a mesma compreensão dos requisitos dos serviços para que um comportamento único fim a fim seja alcançado, independente da quantidade de domínios que estão cooperando para implantá-los ou o tipo de tecnologia de QoS que os domínios estão utilizando. A negociação de serviços se refere ao processo pelo qual os domínios se comunicam para disponibilizar serviços na Internet. Ela consiste em verificar se um serviço pode ser oferecido em um caminho (seqüência de

domínios), em termos das garantias de desempenho que ele necessita. Serviços avançados podem produzir alterações mais freqüentes nas rotas e no volume de tráfego gerado. Portanto, a negociação de serviços deve ser dinâmica, baseada em modelos de negociação eficientes, justos e escaláveis. Em Chameleon, vários modelos de negociação podem ser utilizados, como os modelos cascata, estrela e hierárquico.

Uma avaliação comparativa dos modelos de negociação mencionados acima foi realizada, produzindo conclusões significativas. Os cenários analisados mostraram que o modelo hierárquico é a melhor alternativa para uma grande quantidade de domínios, quando se comparam os critérios de eficiência, justiça e escalabilidade. Além disso, ele é capaz de gerar os incentivos financeiros necessários para a implantação de QoS e a solução dos problemas de interconexão dos domínios da Internet. Quando o número de domínios é pequeno, o modelo cascata se mostrou o mais adequado, devido à sua simplicidade. O modelo estrela é semelhante ao cascata em termos de benefícios oferecidos à negociação, mas ele apresenta problemas de escalabilidade.

To my wife, Nil, and to my kids, Gabriela and Arthur, with love.

Acknowledgments

“I will instruct you and teach you in the way you should go”
(Psalm, 32:8)

“I can do everything through him who gives me strength”
(Philippians, 4:13)

“For everything there is a season, and a time for every matter under heaven”
(Ecclesiastes, 3:1)

I am grateful to everyone who directly or indirectly helped me to complete my thesis. First and foremost, I wish sincerely to thank my supervisor, Prof. Djamel Sadok, who gave me support and guidance during the course of my work. At the beginning of my studies, he introduced me to the area of Quality of Service in the Internet, which is now the subject of my thesis. I think I made a good choice in changing my original research area for the work with Quality of Service. It is much more interesting and has the potential of becoming really useful for the Internet community.

I thank Prof. Judith Kelner, who helped and supported me in many different situations. She gave me the opportunity to stay two months in Berlin/Germany on a research trip, which was very productive for my work in addition to being a wonderful experience. I also thank Michael Smirnov, from Fraunhofer Fokus Institute in Berlin, for our few but very enlightening discussions. He advised me to look into the current problems with the interconnection of domains in the Internet, which resulted in section 2.1 of this document.

I would also like to thank my colleagues (present and past) in the Networking and Telecommunications Research Group (GPRT) of UFPE, who helped me with comments and suggestions to my work during our weekly meetings or in private conversations.

I would like to express my gratitude to my parents, brothers and sisters, for their comprehension and support, mainly for the last ten years that I’ve been living so far way from them, even though at the same country.

Finally, I want to thank my wife Nil, and my kids, Gabriela and Arthur, for their love and patience with the husband and father who could not always give them the attention that they deserved.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
AP	Access Provider
ARMA	Auto-Regressive Moving Average
AS	Autonomous System
ASP	Application Service Provider
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
BB	Bandwidth Broker
BGP	Border Gateway Protocol
BGRP	Border Gateway Reservation Protocol
BR	Border Router
CBR	Constant Bit-Rate
CDG	Chameleon Domain Group
COPS	Common Open Policy Service
CoS	Class of Service
CR	Constraint-based Routing
CR-LSP	Constraint-based Routed – Label Switched Path
DFZ	Default-Free Zone
DiffServ	Differentiated Services
DSCP	DiffServ Codepoint
DWDM	Dense Wavelength-Division Multiplexing
EAT	End-user Application Toolkit

EF	Expedited Forwarding
E-LSP	Explicit Label Switched Path
GIRP	Global Internet Routing Table
GPRS	General Packet Radio Service
HSP	Hosting Service Provider
IANA	Internet Assigned Number Authority
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IPDV	IP Packet Delay Variation
IPPM	IP Performance Metrics
IRR	Internet Routing Registry
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System-to-Intermediate System Protocol
ISP	Internet Service Provider
ISSLL	Integrated Service over Specific Link Layers
ITU-T	Internet Telecommunication Union – Telecommunication Standardization Bureau
LAN	Local Area Network
LDP	Label Distribution Protocol
LS	Local Service
LSP	Label Switched Path
LSR	Label Switched Path
MBAC	Measurement-Based Admission Control
MC	Monitoring Coordinator
MIB	Management Information Base
MMPP	Markov Modulated Poisson Process
MPLS	Multiprotocol Label Switching
MTTR	Maximum Time to Repair
MTU	Maximum Transfer Unit

NAP	Network Access Point
NAT	Network Address Translation
NHLFE	Next-Hop Hop Label Forwarding Entries
NSIS	Next Steps in Signaling
NSP	Network Service Provider
OSPF	Open Shortest-Path First
OWAMP	One Way Active Measurement Protocol
PDB	Per-Domain Behavior
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHB	Per-Hop Behavior
PoP	Point of Presence
PSTN	Public Switched Telephone Network
PTT	Ponto de Troca de Tráfego
QBSS	QBone Scavenger Service
QoS	Quality of Service
QoSR	Quality of Service Routing
RA	Routing Arbiter
RFC	Request For Comments
RIP	Routing Information Protocol
RM	Resource Manager
RRDB	Routing Registry Data Base
RSVP	Resource Reservation Protocol
RTT	Round-Trip Time
SB	Service Broker
SE	Service Exchange
SIMP	Simple Interdomain Monitoring Protocol
SIP	Session Initiation Protocol
SKA	Sender Keeps All

SLA	Service Level Agreement
SLAT	Service Level Agreement Trader
SLO	Service Level Objective
SLS	Service Level Specification
SM	Service Manager
SNMP	Simple Network Management Protocol
SSP	Storage Service Provider
TE	Traffic Engineering
TOS	Type of Service
UMTS	Universal Mobile Telecommunications System
UPN	User Premise Network
USM	User Service Management
VBNS	Very high speed Backbone Network Service
VBR	Variable Bit-Rate
VoD	Video on Demand
VoIP	Voice over IP
VPN	Virtual Private Network
VW	Virtual Wire
WDS	Well-Defined Service
WDSID	Well-Defined Service Identifier
WKS	Well-Known Service
WSP	Wireless Service Provider

Chapter 1

Introduction

This thesis is on the subject of providing advanced services based on quality of service (QoS) guarantees that extend over multiple domains (networks, ISPs) in the Internet. The deployment of advanced QoS-based services is increasingly capturing the attention of the Internet community. QoS is something that has been discussed for some years now, but still there are no technical and commercial solutions strong enough for convincing providers to implement it and users to pay more for services with performance guarantees. QoS guarantees need to be maintained along the entire path between source and destination across several domains often with different administrations and technical characteristics. The big challenge is to develop technologies and business models capable of stimulating domains to offer QoS-based services for source and/or destination users that are outside their networks.

In the following sections, the need of QoS in the Internet and the reasons why it is not commercially available so far are examined. Then, the problem this thesis aims to resolve and an overview of the proposed solution are presented. Following on from this, the assumptions made in the context of this work are also described. Finally, the organization of the next chapters is shown.

1.1 Context and Motivation

The current Internet does not provide any type of performance guarantees to its users. Although users, providers, vendors and researchers agree in the deployment of QoS in the Internet, this is yet to become a reality. When utilizing traditional data applications, users know that at some point in time, if they wait patiently, the desired web page, file or e-mail will reach their desktops. If users cannot wait that long, there is no solution, other than upgrading their connection with the access provider or opting for another one. In certain cases, such as that of home users, even these solutions are not possible, because only slow dial-up connections are available and/or only one access provider exists in the region. Beyond the access provider there is a black box. When utilizing interactive multimedia applications (telephony or videoconferencing) or streaming applications (radio transmissions), users are subject to frequent disruptions and delays that sometimes make it impossible to enjoy them. Under these conditions, neither providers nor users have incentives for turning multimedia applications into reality in the Internet.

Currently, the Internet does not deploy QoS due to the very nature of the best effort service provided by its network layer IP protocol. The best effort service does not make any difference among users and applications, and packets are randomly delayed and discarded at times when the network is congested. In this last decade, a number of solutions have been proposed for extending the best effort service and providing guarantees for selected emerging applications. IntServ, DiffServ and MPLS (section 2.3) are examples of technologies that are currently standardized and available in commercial products. These technologies can be used successfully for deploying QoS in corporate networks and in small portions of the Internet. However, none of them was able to motivate the interested community in deploying end-to-end QoS in the whole Internet. It seems that there are more obstacles other than the best effort service stopping providers from adding meaningful QoS guarantees to their service offerings at an Internet-wide scope.

There is a recent trend for Internet services to be covered by Service Level Agreements (SLAs), which are, basically, contracts between users and providers. A SLA offers some guarantees at a higher level for users at the same time it establishes some punishments and settlement clauses for providers in the case of service outages or malfunctioning. The expected result is that the synergy of combining the power of providing QoS guarantees of the enabling technologies with the assurances and flexibility provided by the SLAs will change the current scenario of Internet QoS. However, most examples of existing SLAs only consider simple guarantees for the best effort service on a single backbone network.

Some restrictions for the applicability of SLAs and implementation of QoS technologies come from the current nature of the interconnection of domains in the Internet. This area is self-organized, i.e., there is no governmental regulation, and so far it has not found stimuli for providing enough financial incentives for the deployment of QoS-based services. The combination of the settlement structures of the two most common forms of interconnection, transit and peering, currently creates a situation where only half of the end-to-end path of a packet is covered by financial compensation. Establishing and negotiating SLAs in such a scenario is very difficult.

Domains maintain several bilateral agreements with each other in order to exchange traffic and routing information. These agreements are statically negotiated and the timescale for renegotiation typically is on the terms of months, depending on the evolution of observed traffic. With the introduction of new QoS-based advanced services, this situation tends to worsen, due to the more frequent changes in service utilization patterns and the urge of finding new routes for meeting services' performance requirements. Therefore, *dynamic* service negotiation becomes extremely necessary so that resources are correctly provisioned along the end-to-end path between traffic source and destination.

The motivation for this thesis comes from the very compelling context for QoS in the Internet described in the above paragraphs. Firstly, the lack of QoS guarantees, contrasted with the great interest of the community. Secondly, the evidence that the best effort service is not the culprit alone, since there are standardized and commercially available QoS technologies. Thirdly, introducing SLAs for Internet services is a promising attempt, but it is far from being a complete solution today. And, fourthly, the interconnection of domains and the structure of the financial settlements among them are also an impediment for the deployment of QoS-based services in the Internet. Therefore, any solution for the QoS problem needs to take into account the above-mentioned issues, while at the same time having practical and feasible ideas.

1.2 Problem Definition and Proposed Solution

The discussions in the previous section provided enough background for defining clearly the problem that this thesis intends to deal with. The main objective is the proposal of an architecture for providing end-to-end QoS-based advanced services in the Internet. The meaning and implications of this sentence are explained below:

- Quality of service (QoS) is defined as both the performance of a network relative to application needs and the set of technologies that enable a network to make performance assurances.
- Service is the treatment provided by routers and other network elements to packets of an application as they traverse the path between source and destination domain. An

advanced service is every service that is able to break the limitations of the best effort service, by offering performance guarantees according to QoS metrics.

- End-to-end means that the communicating users may access the Internet from different administrative domains (network), i.e., they are not restricted to use services within their particular service provider. This distinction is important in the sense that there is a strong trend toward providing QoS guarantees for some intradomain services. One illustrative example is a virtual private network (VPN) for connecting headquarters and branch offices of the same company or creating extranets for e-business among different companies.
- An evolutionary approach is assumed, whereby the current Internet is the basis for a true converged network [61] (delivering data, voice and video). To put it more simply, this proposal does not require the current Internet to be abandoned and a brand new network to be built. The proposal aims at gradually introducing new QoS-based services in the Internet by tackling both problems and limitations of the best effort service and those from the interconnection of domains.
- The Chameleon architecture is proposed as an overlay (virtual) network over the Internet capable of resolving the main issues related to QoS from a higher level of service abstraction, while keeping in mind practical solutions for providing the required QoS guarantees for the applications.

There are many challenges for achieving end-to-end QoS. The main issue here is that deploying advanced services is not a matter that can be solved only by implementing mechanisms for resource reservation or service differentiation at the router level. Integrated Services (IntServ) and Differentiated Services (DiffServ) are the main approaches for QoS in the Internet, currently available in commercial products. IntServ provides support for end-to-end services by its natural design, but it raises scalability concerns, since it relies on per-hop signaling and state maintenance for each individual flow along all the nodes it traverses. DiffServ provides service differentiation for traffic aggregates and therefore it is able to scale to the Internet core. DiffServ also recognizes the need of a SLA between two neighboring domains, in order to permit an advanced service to be extended from one domain to another. In spite of its superior theoretically proven features, up to the present time, DiffServ has not been considerably deployed other than in some testbed networks.

Clearly, the deployment of advanced services involves more than technical decisions. It is also necessary to deal with new issues such as service definition and negotiation. This higher abstraction layer, in the form of an overlay network, is the subject of this thesis. At this level, policies and business models can be more easily combined with QoS technologies for building services that present the right incentives for the end users and for all involved domains and ISPs. Four important aspects should be observed:

1. Domains need to decide which services are worth deploying and how they will agree on the service definition.
2. As traffic volumes vary over different timescales for different services, domains will probably want to renegotiate the service levels of the SLA with each other in a more dynamic and automatic way.
3. It should not be expected that every domain would use the same underlying QoS technology for providing the agreed performance guarantees. The overlay network dealing with service aspects should provide domains with means to choose any QoS technology, at least in theory. Some domains may opt not to use any QoS technology at all, relying solely on an over-provisioned network.
4. A comprehensive understanding of the life cycle of advanced services in the Internet is required, having in mind that any successful solution has to be driven by sound business models and not by the enabling technologies. For instance, along with service definition, negotiation and implementation, other activities such as service monitoring and accounting (pricing, billing, etc.) should be considered.

Most of these aspects are incorporated into the Chameleon architecture that was developed to provide QoS-based advanced end-to-end services in the Internet. It is divided in three logical planes, in order to provide flexibility to service definition and negotiation, efficient implementation and control of proper operation of contracted services, namely: the service, operation and monitoring planes. The service plane creates an abstract model of the network, so that every domain offers a similar external interface for service definition and negotiation. It plays a fundamental role in Chameleon. It can be seen as an overlay plane over the operation plane and its implementation does not cause further changes to the existing infrastructure, except those that are necessary for implementing the service, e.g., introducing a QoS technology. Its main functions are service definition, service negotiation and resource management. In the operation plane domains implement negotiated services through some QoS technology. The monitoring plane is orthogonal to the other planes and its main responsibility is assuring that services are operating within the agreed quality levels.

1.3 Assumptions

The QoS area is extremely broad and no single proposal can deal with all the aspects that it involves. Therefore, this section states assumptions that we made within the development of the Chameleon architecture and also some scope limitations.

- A service refers to a "transport service" (section 2.2.4), which is a service for connectivity such as the best effort service, and not a user service. Unless in some particular cases, no attempt is made in this thesis for elaborating on particular solutions for user services, such as interactive voice and video services. Throughout this document, the explicit distinction between them was made only in those places where the use of the single word "service" could introduce ambiguity.
- No particular IP QoS technology is being considered for implementing the Chameleon architecture, even though in some places there are specific references to some of them. However, there are some obvious recommendations for the use or the best combinations of each technology, as those described in section 3.3.4.
- Service negotiation refers to "dynamic interdomain end-to-end advanced transport service negotiation". The negotiation is dynamic, among various domains and the results of the negotiation are always end-to-end (section 5.4.1), unless otherwise stated.
- The solutions presented in this document refer exclusively to QoS at the IP level[1]. Neither a particular datalink layer technology is envisioned nor any of them is explicitly excluded. The scope of the Chameleon architecture extends wherever there is an IP-enabled computer system, which can be a host, a router, or a gateway to a non-IP network, such as the PSTN (Public Switched Telephone Network). Particular issues regarding the provisioning of QoS in some technologies are not covered, although they are very important for the end-to-end QoS deployment. A number of

[1] Solutions should work for both IPv4 and IPv6 protocols whenever possible, although this is not an assumption for this thesis. It is assumed that all solutions should work in the current Internet, with the IPv4 protocol, and that the gradual introduction of the IPv6 protocol would not pose any problems to the continuity of the QoS deployment, as proposed in this thesis. IPv6 can be used in conjunction with IntServ, DiffServ [150] and MPLS [11].

QoS solutions for specific technologies such as Ethernet, ATM and Frame Relay [55][78], as well as for mobile environments have been proposed [40]. The support of IP QoS is being developed for the new generation wireless technologies such as GPRS (General Packet Radio Service) [166] and UMTS (Universal Mobile Telecommunications System) [114]. Similarly, recent advances in physical layer technologies are not specifically the concern of this thesis despite their role in future networks. For example, network providers are expected to implement in the next few years some form of IP over WDM [91], an enabling technology for the new high bandwidth consuming applications, such as Video on Demand and teleimmersion.

- The solutions presented in this thesis are only valid for unicast traffic. Multicast traffic would require different considerations for resource estimation (section 3.5.1) and maybe different negotiation models to those presented in this work.
- The SLA covers only the scenarios where one source domain is negotiating with one destination domain at a time. This style (or scope type) is called Pipe [65]. Other styles involving more domains, such as Hose and Funnel [93], could also be allowed in Chameleon, but they are out of the scope of this thesis due to their high complexity.
- An end-to-end service does not include the aspects of top-to-bottom QoS (section 2.2.3), which involves operating systems, middleware and applications.
- Service accounting (involving pricing, billing and charging – section 2.2.6), although extremely important for the deployment of QoS is not considered in this thesis.
- Security issues are not covered in this work, although they are known to play a crucial role in a successful implementation.

1.4 Thesis Organization

This thesis is structured for introducing the concepts and proposals in a logical fashion. In this chapter, the problem of end-to-end QoS in the Internet was introduced and the proposed solution was outlined. The remainder of this document is organized as follows. Chapter 2 reviews the background in the Internet QoS area. First, it analyzes the influence of the methods and structures for the interconnection of domains on the lack of QoS in the current Internet. Then, it presents some concepts involving QoS-based services in the Internet, including a proposal for a service life cycle. Finally, it introduces the main technologies that are being developed for deploying QoS in the Internet.

Chapter 3 introduces the Chameleon architecture, elaborating on the service, operation and monitoring planes and their interactions. It also presents the important concept of a Chameleon Domain Group (CDG), which represents a group of domains willing to deploy the Chameleon architecture together with each other. The resource control activities for service provisioning are also described.

Chapter 4 describes the strategy adopted in Chameleon for service definition built on the concept of Well-Defined Services (WDS). The need for explicitly defining services, as opposed to allowing domains to freely configure parameters and values of the Service Level Specifications (SLS) at negotiation time, is explained. The concepts of WDS classes and instances, their format and content of are presented, as well as an elucidating example. This chapter also describes a scenario for the identification of packets that belong to a WDS, as they travel from source to destination, traversing domains that implement different QoS technologies. Finally, the Chameleon's approach service definition is compared to other existing approaches.

Chapter 5 describes the main ideas related to interdomain dynamic service negotiation in Chameleon. The concepts developed for service negotiation are important contributions of this thesis. This chapter first describes the service deployment model, comprised of user and transport service negotiations. Then, the service negotiation process and a simple taxonomy for classification of negotiation models are presented. Next, five negotiation models are presented: the cascade, hub, hierarchical, wave and border models. The hierarchical one is given special attention as it is deemed to bring more benefits for deploying QoS in the Internet. This chapter also gives some directions for the interaction of CDGs, which can use both homogeneous or heterogeneous negotiation models.

Chapter 6 is aimed at evaluating the cascade, hub and hierarchical models, mostly based on a simulation study. The criteria for comparison are efficiency, fairness, scalability, reliability and resilience, financial incentives, and complexity and costs. The hierarchical model is found to be the best alternative, when comparing efficiency, fairness and scalability. It is also able to provide the right financial incentives for deploying QoS and resolves some of the current problems of the interconnection of domains in the Internet.

Chapter 7 concludes the thesis and discusses directions for future work.

Chapter 2

Background

Providing QoS in the Internet has been a topic of great interest for users, providers, vendors and researchers in recent years. The main motivation is turning the Internet into a true converged network, where real-time multimedia applications can be deployed together with the more traditional data applications. However, the goal of achieving end-to-end QoS has to be fulfilled with the current Internet as the starting point, as opposed to the experience of proposing a very compelling technology (ATM), but which required a brand new network did not succeed in the past. The rationale is that simply deploying QoS technologies disregarding the current impediments is naïve. Therefore, this chapter aims at presenting background information on the main problems and solutions in the Internet QoS area and the challenges facing QoS deployment.

In the sections that follow, the author's view of the most important issues regarding Internet QoS is exposed. Section 2.1 discusses how the current adopted models for the interconnection of domains can adversely affect the achievement of QoS in the Internet. In section 2.2, some concepts related to QoS-based advanced services are presented in a higher level of abstraction (with no particular technical solution in mind). In section 2.3 the approaches that have been seriously considered for implementing QoS in the last decade are presented. Finally, section 2.4 summarizes the most important contributions of this chapter for the understanding of the proposals made in the rest of this thesis.

2.1 Interconnection of Domains in the Internet

The Internet is comprised of thousands of networks capable of exchanging traffic by means of the IP protocol. A network with technical and operational autonomy is normally called an administrative domain. Throughout this thesis, the term “domain” will be used, for short. Domains are interconnected to each other, through interdomain links, so that traffic can be forwarded among them. In order to get access to the Internet, a domain needs an interconnection to at least one other domain that is already interconnected.

Most packets in the Internet traverse between three and six domains as they are forwarded from source to destination hosts [160]. For example, packets from a host connected to the domain Inter·Net (www.br.inter.net) in Recife/Brazil to the domain UOL (www.uol.com.br) traverse four different domains: Inter·Net, Telemar (www.telemar.com.br), Embratel (www.embratel.net.br) and UOL. From UFPE/Brazil (www.ufpe.br) to University of Berlin/Germany (www.uni-berlin.de), packets traverse six domains: UFPE, RNP (www.rnp.br), Abilene (www.abilene.edu), GÉANT (www.geant.net), DFN (www.dfn.de) and the University of Berlin. The logical implication of this fact is that all domains along the path from source to destination have to cooperate in order to provide performance assurances for the new advanced QoS-based services.

2.1.1 The Internet Backbone

Apart from being called the network of the networks, there is very little understanding about what really is the Internet and where it is. Any computer that has a connection to the Internet (by any means), with a valid IP address and properly configured with the TCP/IP protocol suite is part of the Internet. However, beyond the Access Provider, the Net is a black box from the user’s point of view. The network infrastructure that permits users to establish communication with each other in a worldwide scope (i.e., in the Global Internet) is called the Internet backbone (or Internet core).

The structure of the Internet backbone has changed radically over the years. Not only the network capacity increased (from 56 Kbps to up to 10 Gbps today), the topology has also changed extensively. The Internet began as the ARPANET in 1969, connecting four sites. It was expanded with the connection of other networks, and the ARPANET served as the backbone for interdomain communication until 1985, when the NSF (National Science Foundation) built a network called NSFNET, due to the worsening congestion situation of the ARPANET.

With the decision of making the Internet available for commercial use, in 1995 the NSF funded a new infrastructure for the Internet backbone, for superseding the NSFNET in carrying the heavy traffic among networks. It consisted of the following features and components:

- vBNS (Very high speed Backbone Network Service) [143]: A network to provide 155 Mbps of bandwidth for interconnecting supercomputing centers, research facilities, and educational institutions. Currently, the vBNS operates with links of up to 2.5 Gbps.
- NAP (Network Access Point): NAP is a place where many domains exchange traffic with each other. It is also called Internet Exchange (section 2.1.4). Initially, four public NAPs were created, in different geographical locations in the USA.
- NSP (Network Service Provider): NSPs are private networks aiming at carrying interdomain traffic and that should be connected to at least three NAPs.
- RA (Routing Arbiter): The RA [69] provides routing information for all domains connected to the NAPs and other Internet exchanges, thus eliminating the need for domains to exchange routing information to any other domain. The RA consists of a Routing Registry Data Base (RRDB), which is a centralized database of routing information, and Route Servers (RSs), which implement the RA functionality.

NAPs and NSPs are the heart of the current Internet backbone. In countries other than the USA, a similar structure comprised on NAPs and NSPs is being built. An interesting fact is that, initially most NSPs worldwide were connected to each other through the NAPs in the USA, since there was no direct interconnection among them. This situation remains true in a number of countries and continents. For instance, from Brazil to most countries in the South America, the traffic is forwarded through NSPs and NAPs in the USA. The implication is that the traffic traverses a much longer path between source and destination, thus increasing the chance of crossing congested points and at the same time contributing for the congestion.

2.1.2 Types of Domains in the Internet

Domains can be classified according to different dimensions in the Internet. One of them refers to whether it is a network used only for access by particular users (User Premises Network - UPN) or a network that is shared by UPNs for exchanging traffic between them. The latter part is called the Public Internet. An UPN can vary from just one computer (home or mobile user) to a large enterprise with thousands of computers. User networks are connected to the Internet through access networks, which can be based on different technologies, such as dial-up connections, ISDN, xDSL, cable modem, GPRS and some form of dedicated circuits (E1, E3, Frame Relay, ATM). They are sometimes also called "stub" networks.

An Internet Service Provider (ISP) is a general reference for any domain that provides some type of service for users[2] in the Internet, from basic interconnection to specialized content distribution. Some common types of providers (xSPs) are[3]:

- ASP: Application Service Providers offer users access to software applications and related support services, such as e-commerce, e-mail and gaming.
- HSP: Hosting Service Providers or more commonly, Web Hosts, are companies that make disk space and services available for users to host web-based content.
- SSP: Storage Service Providers are xSPs that offer network based storage for companies that wish to outsource their data storage and management.
- WSP: Wireless Service Providers offer users fixed and mobile wireless access to the Internet. It is expected that in the next few years, more than 30% of the users will access the Internet through their mobile phones or palmtops [75].
- NSP: Network Service Providers are comprised of telecommunications companies and data carriers that interconnect the other types of providers. They are also called backbone providers or transit providers.

ISPs can be also categorized into tiers, depending on the size of their network and number of subscriber (more used in the USA). According to this view, there are three tiers of ISPs:

- **Tier-1** (global or national): These are very big providers, with nationwide networks and over 1 million subscribers. It is considered that there are only about ten Tier-1 ISPs in the USA [45]. The concept of Tier-1 ISP is somewhat similar to the concept of NSP, although the latter does not need to have its own subscribers. Tier-1 ISPs also have access to the global Internet routing table (section 2.1.6) and do not purchase transit (section 2.1.5) from anyone.
- **Tier-2** (regional): Tier-2 ISPs usually own a regional network and support over 50.000 users. They can offer nationwide services by interconnecting to Tier-1 ISPs.

[2] From this point on, users refer to both individuals and enterprises.

[3] According to XSPsite.net (www.xspsite.net).

- **Tier-3** (local): These are the most common type of ISPs in the Internet, offering only local services.

2.1.3 Interconnection Structure

Independently from the type and function of a domain, it has to be connected to other domains in order to be effectively useful for its users. There is no such a thing as a single Internet topology, because it is formed by a not well organized mesh of interconnected global, national, regional and local domains. In most cases, simply being connected to a domain that is already connected is enough for guaranteeing full interoperability with every other domain in the Internet. However, in some cases, this is not true, for example in the case of the recent bankruptcy of one major Tier-1 ISP in July 2001, when some users could not reach some destinations in the Internet during quite a long time [5]. This is not the expected situation, though.

The simplest interconnection model for the Internet is a purely hierarchical one, as depicted in Figure 2.1. In this model, local ISPs (Tier-3) connect to Regional ISPs (Tier-2) that connect to National ISPs (Tier-1), which in turn are connected to each other. Three different scenarios are represented in Figure 2.1. User E and user F are subscribed to the same local ISP. Therefore, they generate local traffic only. Users A and B are subscribed to different local ISPs, which are connected to the same regional ISP. In this case, their traffic traverses a broader area, being forwarded by a higher number of routers. The extreme case is that of users C and D, where the traffic is forwarded up to the national ISP, traversing the NAP and another national ISP before it is forwarded to the destination, through the regional ISP. Even though both users live in the same city, traffic between them can travel thousands of kilometers.

The main problem with the purely hierarchical interconnection model is that it is bad for the resource utilization of the ISPs, since the shorter the path is, the less resources are used. In other words, shorter paths are more efficient and cheaper. Longer paths also harm the quality of service for users, because then the traffic is subject to higher delays, packet losses and consequently, lower throughput. Therefore, the direct interconnection of national and regional ISPs is becoming increasingly common, as shown by the dashed horizontal lines in Figure 2.1. Even local providers and corporate networks are installing more and more connections to the Internet, mainly for the purpose of resilience.

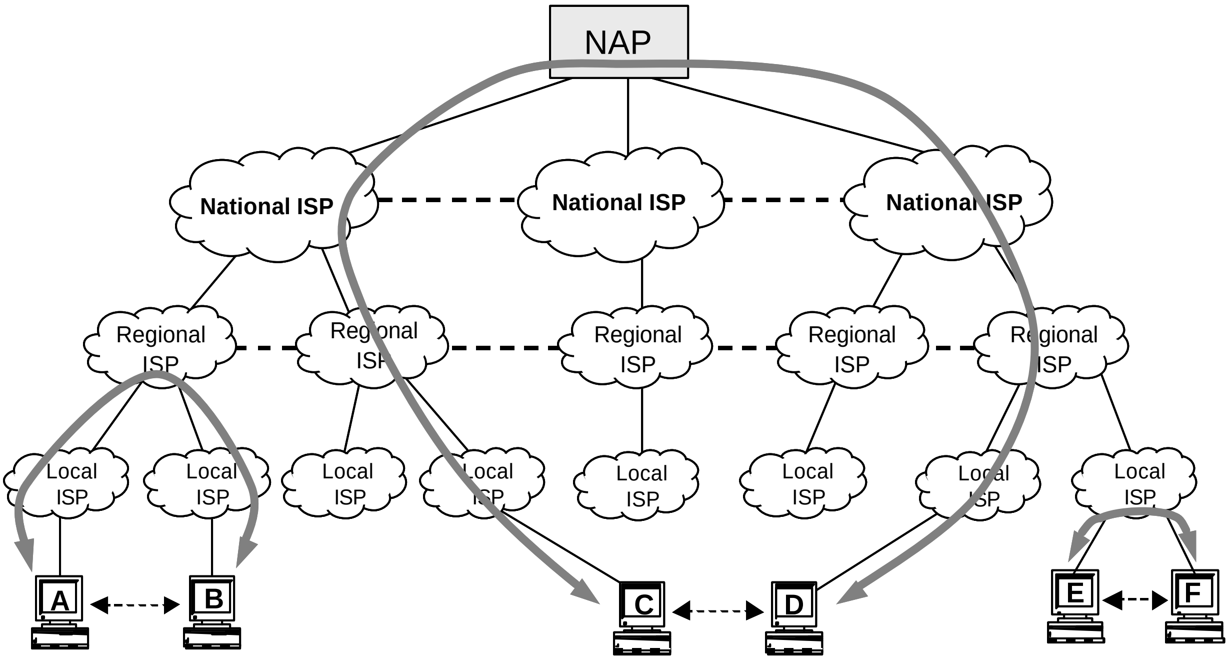


Figure 2.1 – Hierarchical interconnections of ISPs in the Internet

2.1.4 Interconnection Models

The interconnection involves the provision of some bandwidth among the interconnecting parties, which can be done by two different approaches: exchange-based or direct circuit interconnection [154]. The exchange-based model follows the idea of the NAP (or Internet Exchange), that is, it is seen as a special provider-like entity where various ISPs have interconnection links. At this point, domains can exchange routing information and packets with each other, depending on the type of business models they adopted and the agreements among them.

The most common implementation for an Internet Exchange is by means of a switched facility. Each ISP places a router at the exchange building and provides a dedicated link between its network and the exchange. The routers are connected to each other through a link-layer switched technology, usually Ethernet, but also FDDI and ATM are used. Whether a router needs to exchange routing information with every other router in the exchange, depends on the availability of a Route Server at the exchange and the willingness of the ISP for using it (section 2.1.1).

The original NAPs were public exchange points, so that ISPs did not need to pay for simply connecting to them. In the last years, other countries also have built their NAPs following the structure of the USA. In Brazil, there is one NAP (PTT – Ponto de Troca de

Tráfego), managed by the ANSP network, currently with 32 participating domains [71]. Private NAPs as well as exchanges offering a variety of value-added services also have been emerging, such as Internap [111] and Equinix [63].

The direct circuit interconnection model is easier to be implemented, since it does not depend on the existence of an Internet exchange at the particular location where two domains want to exchange traffic. A direct circuit is a point-to-point link between to ISPs, provided by a telecommunication carrier, based on SDH (e.g., E1 and E3 links), Frame Relay or ATM. In order to avoid congested NAPs, most national ISPs are relying on direct circuits for their interconnections.

2.1.5 Business Models

Business models refer to the commercial relationship that governs the traffic exchange among two or more domains as well as to the settlement structures that permit the cost of this service to be shared among domains. Two types of relationships are more common in today's Internet [105][155]: transit and peering. There is also a third one that is found in some particular cases, known as the sibling relationship [88]. All the three models can be implemented by means of an exchange-based or a direct circuit interconnection.

Transit is a business relationship whereby one ISP sells access to all destinations in its routing table, i.e., supposedly to the whole Internet. In a simple world, local ISPs buy transit from regional ISPs that in turn buy transit from national ISPs. The latter do not buy transit capacity from any other ISP, because they are considered peers by each other. Peering is a business relationship whereby ISPs reciprocally provide access to each other's customers. In a peering relationship, there is no payment for the interconnection, because the involved ISPs are considered by each other to have the same size, thus submitting similar traffic volumes. Tier-1 ISPs seek to peering relationships with as many other peers as possible, broadly for technical reasons, rather than for reducing the cost of transit, since they do not purchase transit from any other ISP. Peering has the benefit of lower latency, better control over routing, and may therefore lead to lower packet loss. In other words, peering may be used for improving the quality of service provided to traffic.

In practice, the situation is far more complicated, because different ISPs have different and incompatible points of view about whether they are clients (or providers) or peers. Big ISPs want to have as many clients as possible, so that the high costs of their national and global networks can be compensated with higher revenues. On the other hand, smaller ISPs want to have as many peers as possible, in order to lower their interconnection costs. The problem worsened because initially in the new Internet backbone (section 2.1.1), some of the big national providers used to accept to be peers of much smaller regional or even local ISPs. At some point, most of them started to demand those smaller ISPs to be clients in a transit relationship (i.e., involving payments) [57]. At present, the rules for accepting peering relationships are usually very strict [155].

A sibling relationship happens among domains that do not see each other as a competitor. Universities located at a same city, which do not have financial resources sufficient for buying high-speed connections, usually share their Internet connections, so that they can be benefited from each other's idle capacity. A similar situation happens with companies belonging to a same organization (a holding), where a transit relationship may not make sense.

Financial settlements have been a continuous topic of discussions within the area of Internet interconnections [106]. Similar to any other multi-provider distributed service, Internet access is faced with the issue of cost distribution among providers. The access provider is the ISP that collects the money from the users, which will be used for moving packets to and from the whole Internet. Since there may be several domains along the end-to-end path between source and destination, this money should be distributed to other domains.

In the current Internet, however, this issue of financial settlements is not well resolved. Two models are commonly used in conjunction, based on the transit and peering business relationships [105]. The first model is known as the customer-provider one, whereby the user pays its access provider, which pays for its upstream provider and so on. The second model is the Sender Keeps All (SKA), which is a form of relationship without any form of financial settlement.

In any case, a domain is not directly paying for its packets to traverse the end-to-end path. Either it pays for sending all its traffic to (and receiving from) the upstream provider or it does not pay anything for sending or receiving packets. In both cases, the sending domain (and ultimately, the user) expects its traffic to reach the destination. However, it has no idea of the whole cost of delivering its traffic to any destination in the world that is covered by the Internet. This is the reason why in Internet users pay a flat fee for exchanging data with any other location, unlike the telephony business model.

2.1.6 Interdomain Routing

As a packet is forwarded from source to destination in the Internet, each router makes an individual routing decision. Routers pick the IP destination address from the packet, look up into its routing table and choose the next-hop whereto the packet will be forwarded. In order to take correct routing decisions, each router needs to have a coherent view of the network topology. Routing protocols are in charge of exchanging routing information, so that routing tables at each router are updated as soon as the network topology changes. Since the Internet is very large, it is comprised of a large number of routers. A flat network structure, where every router knows the entire Internet topology, would need thousands of routers exchanging routing messages with each other. Therefore, the Internet deploys what looks similar to a hierarchical routing and the unit of routing is called Autonomous System (AS) [104]. According to RFC 1930 [101], “An AS is a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy”. For the sake of simplicity, in this thesis a domain is also an AS.

Routing in the Internet is divided up into two different levels: interior (or intradomain) routing happens within each AS, whereas exterior (or interdomain) routing is used for exchanging routing information among ASs. For intradomain routing, any routing protocol can be used, most commonly OSPF (Open Shortest-Path First) and RIP (Routing Information Protocol) are considered. As far as interdomain routing is concerned, it does not matter which particular intradomain protocol is used, as long as it generates correct routing information.

For interdomain routing, the current de facto standard is the BGP-4 (Border Gateway Protocol 4) [168]. BGP runs in every border router (a router that is connected to another domain). In order for two domains to exchange routing information, both have to consider each other as a peer. Thus, the relevance of the routing servers in Internet Exchanges is clear, as discussed in section 2.1.1. Domains need only to establish BGP peering sessions with the route server. Otherwise, each domain needs to establish peering sessions with every other domain in the exchange. In every AS, border routers run both the intradomain protocol and the BGP. They aggregate internal routing information and pass it to their BGP peers. They also receive routes from their peers and pass them on to the intradomain protocol.

By means of exchanging BGP messages with their peers, domains can gain access to the Global Internet Routing Table (GIRT), which contains routes to every network connected to the Internet. Usually, only Tier-1 ISPs need to have access to the GIRT, because they do not buy network capacity from any other ISP. Smaller ISPs receive partial views of the GIRT, and find the other missing networks by means of a pre-defined “default" route. When looking up an IP address into its routing table, a router chooses the default route when no other specific route is found. Tier-1 ISPs do not use default routes. As a result, the Internet backbone is sometimes called the Default-Free Zone (DFZ) [73].

Unlike intradomain routing, interdomain routing is not solely based on technical information, but also on pre-defined policies. Thus, ASs can opt for selectively revealing their route tables, for different reasons, such as political or business ones. This facility can cause some limitations, though, as described in section 2.1.7.

2.1.7 Problems with the Interconnection of Domains

The interconnection of domains remains an open issue to the present time. There are problems that create barriers for delivering the required quality of service for applications that demand additional guarantees. Peering relationships are at the heart of those interconnection problems in the Internet, whereas transit relationships, NAPs and the settlements structures are also contributing factors.

The problem with public NAPs is that they are always congested, since there is no charge for attaching to them and they offer the possibility to establish transit or peering relationships with several major ISPs. There is also no central entity with the right financial incentives for resolving the congestion problem. Even private NAPs have problems, since they charge on a link capacity basis. Therefore, they do not have financial incentives for upgrading their facilities for coping with increasing traffic volumes, since they cannot charge more until domains decide to upgrade their connection links.

Peering once was thought of as the solution for escaping from congested public NAPs. However, this solution turned into one of the main causes of congestion in the Internet. The main difficulty with peering is that there is no payment involved in the traffic exchange between peers. Due to the operational cost associated with the network infrastructure (there are relatively high costs involved in its installation and maintenance), not paying for forwarding traffic creates a somewhat vulnerable scenario. The most common reported problems are [3][5][12] [89][105][113][155]:

- Lack of financial incentives: There are not financial incentives for upgrading the capacity of the peering point (exchange-based or direct circuit). Since the peer is also a competitor, by upgrading the peering capacity for sending more traffic, a domain is also improving the quality of the peer's networks. The result is that peering is considered one of the main causes of congestions in the Internet, although this fact cannot be proven because domains do not reveal statistics of their peering points.

- Long renegotiation timescales: Due to the lack of financial incentives, the capacity of the peering points usually lasts too much time before being upgraded.
- Routing inefficiency: Because major Tier-1 ISPs do not pay for access to each other's network, the bulk of traffic on any given network is going to and from non-paying customers. To avoid the high costs associated to upgrading their network, domains engineer their routing to get rid of the traffic as soon as possible, creating a routing style called "hot-potato". Hot-potato routing causes packets to travel over longer paths, thus potentially harming the quality of service in the end-to-end path. In other words, peering traffic has a lower "status" (priority) than transit traffic.
- Traffic asymmetry: At peering points, frequently one peer transmits more traffic than the other one. This asymmetry is more advantageous for the peer that is submitting more traffic. Consequently, the other peer does not have much interest in upgrading the peering point, since it can be overloaded with even higher traffic volumes.
- Partial routes: In a peering relationship, domains do not reveal their full routing table, to the use of their network as transit by their peers. For instance, let us suppose that domain B peers with domain A and domain C and receives routes from both. Domain B usually does not reveal routes learned from A to C and vice-versa, so that it does not forward non-paying traffic through its network, even though it is the shortest path between A and C.
- Lack of an SLA (Service Level Agreement): An SLA is a contract that states some guarantees for the customer (section 2.2.5). Since in a peering relationship there are no customer and provider roles, there is consequently no SLA.

The current deployed financial settlement structures, derived from the business relationships, can also be seen as the cause of the lack of quality of service in the Internet. The point is that the combination of customer-provider and SKA models does not remunerate the complete end-to-end path. The fee charged to the customer covers only the part of the network where there is a transit relationship. Once the traffic crosses the first time a peering point, its status in automatically downgraded, due to the lack of payment. A good solution for the interconnection of domains in the Internet must necessarily pass through covering the entire cost of the traffic in the end-to-end path.

Therefore, the most reliable configuration would be a transit relationship with a direct circuit link. This is not always an adequate solution, though, because transit is usually very expensive and big providers will probably refuse purchasing transit from each other. Another solution could be a sort of usage-based billing. Unfortunately, it is not that simple. The difficulty is in determining who has to pay for the traffic: the sender or the receiver. In some cases, the sender is the most interested part, for example, when it is sending an e-mail message, whereas in other cases, such as web browsing and file transfer, the receiver is the interested party. Therefore, a solution that takes into account the interest in data transmission is needed.

In the current scenario, improving the QoS in the Internet is something very difficult to be achieved, because it does not provide the rights incentives. New technical solutions and business models need to be developed to deal with many unanswered issues [89].

2.2 QoS-based Advanced Services in the Internet

In the last decade, the Internet has evolved from an academic and research network to an infrastructure that supports many commercial applications. Now, market forces are pushing the Internet to deploy a new breed of applications, such as interactive audio and video, which require stricter performance guarantees that the Internet is currently unable to provide. In section 2.1 it was stated that the current structure of the interconnection of domains in the Internet is one reason behind its poor service quality. Other reasons are the nature of IP's best effort service (section 2.2.1) and the lack of a comprehensive structure for dealing with service in a higher level of abstraction, i.e., as an overlay network.

A QoS-based advanced service (or simply advanced service, for short) is defined as every service that goes beyond the basic best effort service, allowing the network to offer some performance guarantees to the users. To this end, it needs a well-defined service model, even though it may offer a lower performance than the best effort service. Another name for an advanced service may be "QoS-enabled service". Similarly, a network that deploys advanced services may be called a "QoS-enabled network".

A service model may be described as "an abstract definition of the service that the network client will receive [138]". It specifies a long-term contract between the network and the application designer by defining a stable interface, whose details may change, but the semantics of the service cannot. Thus, the service model documents the commitments that the network makes to a set of clients when they request that service. In other words, a service model is a complete specification of the behaviour that users will expect from a service. The deployment of new advanced applications in the Internet (e.g., interactive audio or video) depends on these well-defined service models.

2.2.1 The Best Effort Service

There is no formal service model definition for the default best effort service provided by the IP protocol. RFC 1812 [23] states that the IP protocol provides a connectionless service, with no end-to-end delivery guarantees. Packets may arrive at the destination host damaged, duplicated, out of order, or not arrive at all. The delay requirements for the best effort service are ASAP (as soon as possible) and the throughput bandwidth requirements are AMAP (as much as possible). Applications can use packets as soon as they arrive and send as much packets as possible, given the current network conditions.

The best effort service applies equal treatment to all users and applications. When a packet arrives at a router, it chooses the next hop and puts it in the packet queue associated with the selected output interface. The default type of a queue used by most routers in the Internet is based on FIFO (First In First Out) scheduling and DropTail queue management. In other words, an incoming packet is always put in the end of the queue. When the queue is full, the packet is simply discarded (dropped), regardless of the user or the application that sent it. Although more sophisticated queue management schemes [32][46][77][134] are available in commercial routers, they are rarely used in practice. As there is no comprehensive solution for deploying QoS in a large network, the higher processing burden imposed by the more complex algorithms are not worth the envisioned benefits.

The rationale for the name best effort is that the network makes its "best effort" to deliver packets ASAP and using AMAP bandwidth. However, congested routers may delay packets for some time or simply discard them when their queues get full.

2.2.2 The Need for Quality of Service

There are several different definitions of Quality of Service (QoS). Everyone who uses the Internet knows that it does not deploy QoS, but currently there is no comprehensive and unique definition of QoS [78] .The definition adopted here characterizes QoS according to two different aspects [193]:

1. The performance of a network relative to application needs.
2. The set of technologies that enable a network to make performance assurances.

This characterization distinguishes the service interface provided by a network to the users of an advanced service, out of the technological choices a network make in order to implement this service. This is particularly useful when services are to be deployed cooperatively by a group of networks that implement different QoS technologies.

With respect to the level of guarantees a network is able to provide, QoS can be broadly classified into two types: resource reservation and prioritization. QoS based on resource reservation offers guarantees for each flow individually, for instance in IntServ (section 2.3.1). This would be ideal for end-to-end QoS, but it raises serious concerns with respect to scalability. Prioritization refers to giving a differentiated treatment to some classes of service (CoS), comprised of traffic aggregates, for example in DiffServ (section 2.3.2). With prioritization, there are no individual guarantees, but the scalability is enhanced.

An important issue to note is that QoS does not create bandwidth [187]. It is not possible to provide what the network does not have, thus making bandwidth availability an important starting point. QoS only manages bandwidth according to application demands and network management settings. This raises the question of whether specific technologies for QoS are really needed or simply by over-provisioning bandwidth the same assurances can be obtained (see discussions in sections 2.3.6 and 3.6). In any case, either using over-provisioning or QoS technologies, some advanced applications need QoS guarantees in order to be effective for their users.

Some driving forces that are leading to the QoS deployment in the Internet are users, ISPs, telecommunication operators and equipment vendors. Users want QoS-based services for use by different applications, such as interactive voice and video, which face performance problems in the current Internet. Most Internet users would accept to be charged on a service basis, if end-to-end performance guarantees were adequately offered. ISPs want QoS as a means for offering value-added services, in order to improve their profits [220]. Telecommunication companies that have been operating two different networks (PSTN and data networks) are willing to deliver both voice and data over a single converged network, in order to save the costs of maintaining a duplicate structure. Most vendors of equipments for the Internet have solutions for deploying QoS according to the current standards (section 2.3) and want QoS solutions to be disseminated in the Internet, in order to increase their profits and market share.

2.2.3 The Challenge of End-to-End QoS

The major challenge in the Internet is to provide end-to-end QoS guarantees, as opposed to only covering part of the path packets have to follow from one communicating host to the another. As end-users see QoS through applications, the notion of QoS guarantees needs to encompass the path from one application all the way to the other application. In other words, both the end-to-end network path between communicating hosts and the top-to-bottom and bottom-to-top paths must be considered in order to guarantee end-to-end QoS.

Figure 2.2 depicts the scenario of end-to-end and top-to-bottom QoS [186]. By the end-to-end perspective, there are IP QoS technologies (such as IntServ, DiffServ and MPLS – section 2.3) being deployed that possibly have to interact with each other, since domains are free to make different choices for implementing QoS. These IP level QoS technologies have to interact with datalink technologies, such as Ethernet, ISDN, ADSL, ATM, Frame Relay, Wireless LANs and GPRS. One of the main features of the IP protocol is accepting to be mapped into almost any datalink technology. However, each technology has its own characteristics that have to be considered individually for providing QoS. Some technologies have built-in QoS features, such as ATM. Some of them are characterized by almost error-free links and high speeds, whereas others inherently suffer from scarce resources and error-prone links. The latter is the case of the wireless technologies, which must be fully integrated in an advanced services deployment scenario, since in the future many users will access the Internet from mobile wireless devices.

From the top-to-bottom perspective, it is not sufficient to extend QoS guarantees to the datalink layer interface at the end systems. Operating systems and applications have to be QoS-enabled in order to make use of the end-to-end guarantees offered by the network. For example, if a given operating system preempts advanced applications even for short time periods for processing some non-priority background applications, the whole effort made by the network can be lost. Another important point is that the application itself should be QoS-enabled, in order to make use of the available guarantees. In other words, the application should be written with QoS in mind, by means of a QoS API provided by a specific QoS middleware. The problem of top-to-bottom QoS is typically tackled by QoS architectures [17].

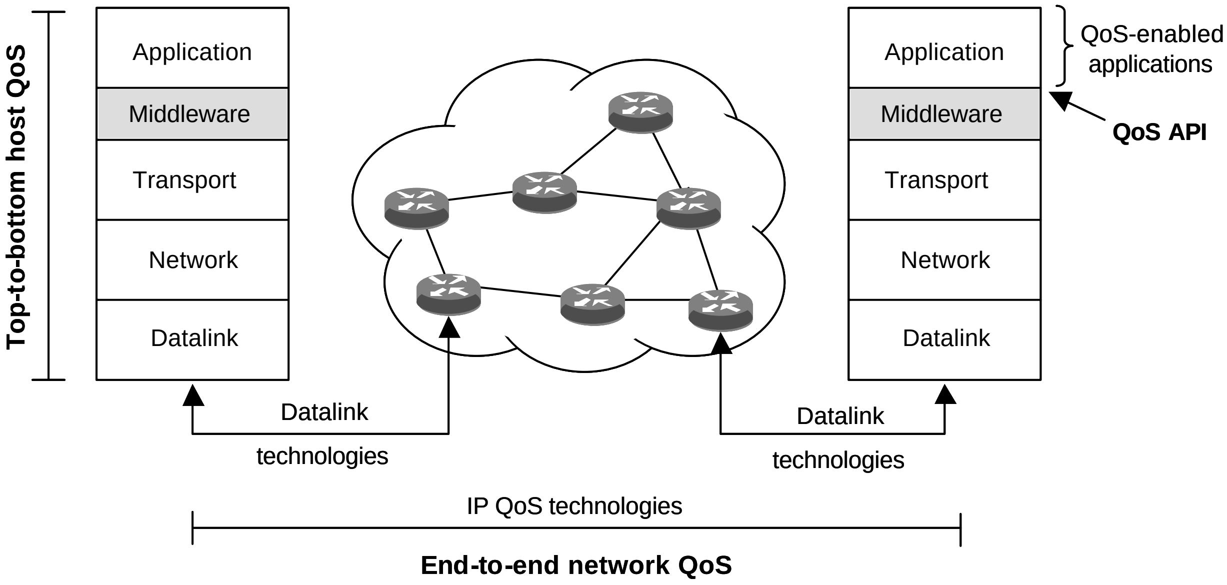


Figure 2.2 – End-to-end and top-to-bottom QoS

This thesis only deals with the problem of end-to-end QoS and even so, within a limited scope, as described in section 3.1.3.

2.2.4 Terminology for Advanced Services

Service is a word broadly used in computer networks to represent the situation where a provider offers something to a client. The nature of the relationship of providers and servers is such that they may be located at any place: at the same process, host, or network. This section is aimed at clarifying the terminology for advanced services that has emerged in the last decade, also used throughput this thesis.

Transport services make up the necessary infrastructure for deploying end-to-end services, which are implemented and negotiated by domains. Examples of transport services are the best effort service, the Leased Line Emulation Service [25][115] and the Assured Service [25][151], which is a “better than best effort service”. End-user services (or just user services, for short) are those services that are meaningful to end-users and are related to their network requirements. Providers that sell services to end-users must map end-user services into transport services in order to be able to participate in the deployment of an end-to-end service. Some examples of end-user services are voice over IP (VoIP) and video on demand (VoD). Transport and user services are sometimes called wholesale and retail services [185], respectively.

With the convergence of the Internet into a true multiservice network, several user services will be offered, which can be further classified in two categories. Communication services are those where the user service provider offers QoS guarantees for the service, but the information (content) that will be conveyed is generated by the user. Examples of communication services are Internet telephony, videoconferencing and VPN (virtual private network). On the other hand, content services are those where the provider owns the content that has some interest to the end-users. A content service is in fact a value-added service, combining a communication services with particular content.

Quantitative services provide concrete guarantees that could be verified by suitable measurements. On the other hand, qualitative services [25] offer assurances that can only be verified by comparison. An example of a qualitative service is “traffic delivered at service class X will be delivered with low delay and low packet loss”. Qualitative services always offer relative guarantees, whereas a quantitative service can offer both relative and absolute guarantees. An example of relative quantitative service is “service class Y will be allotted twice the bandwidth of service class Z”. An example of an absolute quantitative service is “the packet loss rate for service class A will be at most 0.1”

Services may be also classified according to their geographic scope. An intra-domain service is to be used only within the boundaries of a domain, which is a network, e.g., a user network, an ISP or an autonomous system. An end-to-end service may be used when data sources and destinations are located in distinct domains, possibly with several other domains along the path between them.

Guaranteed and predictive services [48] are two broad categories of advanced services that are suitable for supporting multimedia (or playback) applications. In a guaranteed service, each flow (a service client) receives absolute assurances that some a priori performance bounds will be guaranteed, as long as it is conforming to its traffic characterization. The network provides this level of QoS regardless of the behaviour of other flows. For a guaranteed service, the playback point is predefined by the application, so that the network must assure that all packets arrive at the destination before a particular delay bound. In a predictive service, a network controls delay bounds and applications must be able to readjust their playback points according to the current network conditions and have to tolerate infrequent service disruptions. It is assumed that network conditions will remain relatively stable, although the performance of a flow is influenced by the behaviour of other flows sharing the same network segment. Hence, for a predicted service, the network tries to deliver consistent levels of service to its users, but it does not need to compute a priori delay bounds for the worst-case. This implies that the network may achieve higher levels of utilization. Thus, the trade-off in choosing between guaranteed and predictive service depends on the level of QoS that applications need and the cost of providing such a service.

Guaranteed services can be further classified in deterministic and statistical services [79]. A deterministic service guarantees worst-case end-to-end delay bounds for traffic, but it is known to lead to an inefficient use of network resources. A statistical service makes guarantees of the form $\Pr[Delay > X] < e$, where X is an upper bound for the delay and e is the probability of violation of this bound. Thus, it is a service that allows a small fraction of traffic to violate its QoS specifications while it can significantly increase the achievable utilization of network resources. Taking advantage of the statistical properties of traffic, a statistical service can exploit statistical multiplexing gain [140].

Finally, a Well-Defined Service (WDS) is a service that has a clear and unambiguous definition of the performance guarantees that a provider offers or wants to receive when an agreement is being negotiated. It must have the same behavior in every domain where it is implemented, in order to make it possible to deploy an end-to-end service to end-users

2.2.5 SLA and SLS

The implementation of advanced services requires a differentiated treatment for a certain group of packets, which increases the complexity and costs of the network. This cost is obviously shared with the users of the service. Since the user will pay more than when simply using the best effort service, he/she will also require some form of assurance that the service levels will be consistently maintained during the schedule of the subscribed service, through some sort of contract. These assurances are commonly referred to as Service Level Agreement (SLA) and Service Level Specification (SLS).

In a broader sense, a SLA is a formal definition of the commercial relationship that exists between two organizations [212]. Particularly in the networking area, SLA is a service contract between a customer and a service provider that specifies the forwarding treatment a customer should receive [28]. A customer can be an end-user or another provider. A user SLA is a contract between a user and a provider, regarding a user service. On the other hand, a transport SLA is a contract between two providers, regarding a transport service. Thus, an end-to-end service is often formed by the concatenation of a series of SLAs, which together offer the guarantees for the end-users. The user, obviously, does not need to know about how many domains and SLAs have to be signed as a result of its user SLA.

In addition to describing the type and nature of the service to be provided, a SLA usually contains information about a comprehensive set of aspects regarding service guarantees and the rights and obligations of both involved parties [212]. This information can be divided up into three sections [210]:

- Formal section: Legal obligations and rights are described in this part of the SLA. It also defines aspects of pricing and credits, charges or other penalties applicable to a service provider not meeting its obligation (i.e., failing to provide the agreed-upon service levels).
- Customer care section: This section contains aspects regarding customer care, such as the process of reporting problems with the service (who should be contacted, how and when) and the time frame for expecting a response and problem resolution.
- Technical section: The technical section describes the expected performance level of the service and the process of monitoring and reporting the service levels. This part is frequently referred to as the SLS. Formally, a SLS is a set of parameters and their values that define the service offered to a traffic stream by a service provider [93][98][167][175]. The term SLS has been traditionally used in the context of DiffServ (section 2.3.2), but it can be extended to define the service in a higher level of abstraction, regardless for a particular QoS technology. An example of a SLS format and content is shown in section 4.3.1.

SLAs are already a reality in the Internet realm, enhancing the basic best effort service with simple performance guarantees. These guarantees mostly refer to service availability and the monthly average of two-way delay and packet loss[4]. SLAs are provided mainly by large ISPs such as WorldCom[5], Cable & Wireless[6], Sprint[7], and also Embratel[8], in Brazil. Actually, the problem with such SLAs is that they are generally restricted to the provider's network. These big ISPs are merely offering something that they have as a side effect of keeping their networks over-provisioned. Other difficulty for the user is that he/she has to rely on the provider's measurements, that is, frequently it is very difficult to be sure of the performance guarantees the user is paying for [50]. Therefore, it is important for the user to know whether

[4] Two-way delay and packet loss do no reflect the reality accordingly, since they do not differentiate between the forward and backward path, which can contribute in significant different ways for the collected values.

[5] http://www1.worldcom.com/uunet/terms/sla.

[6] http://sla.cw.net.

[7] http://www.sprintworldwide.com/solutions/sla.

[8] http://www.embratel.net.br/internet/info/gd-programa.html.

the SLA measurements are carried out by third-party specialized companies, such as Keynote (www.keynote.com). Nonetheless, recently there has been a tendency for offering multiprovider SLAs [6], that is, a SLA that extends its guarantees to other carefully chosen networks.

2.2.6 Service Life Cycle

In order for an advanced service to be available to end-users, a number of different activities must be fulfilled, that are normally not necessary for the best effort service. A proposal for a service life cycle model, depicted in Figure 2.3, is comprised of the following phases:

- Service definition: Service definition refers to the activity of precisely explaining the semantics of a service in a formal way, generally by means of a SLS (Chapter 4). Domains must agree on the service they will deploy, in order that end-to-end semantics and QoS guarantees can be met. This definition refers to transport services, although user services can also have a precise definition.
- Service implementation: Once a domain knows which services it wants to deploy, it chooses an adequate QoS technology (or a combination of technologies) for implementing them. Different domains, which together deploy an end-to-end service, can use different QoS technologies. Section 3.3 deals with aspects of service implementation.
- Service negotiation: Service negotiation refers to the process whereby a costumer communicates with the provider in order to deploy services in the Internet. A consumer may be a user or another provider. In both cases, it involves verifying the feasibility of deploying a given service through some path (sequence of domains) in terms of its required performance guarantees (Chapter 5).
- Service provisioning: As a consequence of the service negotiation, each domain needs to provision resources according to the adopted QoS technology. Resource provisioning refers to the determination and allocation of resources needed to implement services at various points of the network (section 3.5.3).
- Service utilization: From the user's point of view, the negotiation process encompasses two phases: service subscription and activation (or invocation) [201]. Service subscription is the process of negotiating the right to invoke the service at

some time in the future. Possible results of a service subscription request are permission or refusal. A partial permission is also possible, if there are not enough resources. Service activation refers to the process of actually requesting an amount of resources for using the service and it can be subject to an admission control mechanism. Results of a service activation request are admission or blocking. With some user services, these two phases effectively happens separately, for example, with interactive multimedia services. With other services (such as data services, e.g. a VPN) the service is activated at the time of the service subscription.

Service utilization generates traffic at the network that, in turn, can lead to a new renegotiation and provisioning, in order to better adapt to different traffic patterns and volumes generated by the user applications. Therefore, as long as the service is active, there is a constant cycle between service utilization and negotiation.

- Service creation: User service utilization implies service creation, i.e., it requires all service logic, data and the associated management to be configured at network entities [185]. Apart from configuring routers, which is performed in the provisioning phase, many other entities must be configured in order for an end-to-end service to be available, such as database access and middleboxes [39]. Examples of the latter are firewalls, network address translators (NATs), tunnels, caches and proxies. For instance, there is no advantage in having an end-to-end service configured service through a long path if packets are dropped in the destination firewall because of a misconfiguration. Therefore, services should be created by means of an automatic process that takes into account all network entities in the entire end-to-end path.

  There is a cycle between service utilization and creation. Once the user deactivates the service, it should be torn down. Later on, the user can again utilize the service, that must be created once again and so forth.

- Service accounting: Internet service accounting is defined here as the process of collecting, interpreting, and reporting information related to the cost of the service usage. This process is divided up into four subprocesses [165]: metering, pricing, charging and billing. Metering is the process of measuring and collecting resource usage information. It can be used in both contexts of service accounting and monitoring. Pricing is the determination of a cost per unit. Setting up prices for network services is primarily a marketing and strategic decision rather than a technical concern. The charging process translates the pricing policies into an amount of money that the user has to pay for the resource consumption in service usage. This amount of money is then used by the billing process to inform and bill the user. A proper service accounting strategy is paramount for the success of any service deployment.

- Service monitoring: This phase refers to the process of measuring the actual service levels in order to verify whether they are adherent to the SLS (section 3.4). Depending on the results of the monitoring phase (mainly if the QoS targets are not being met), it leads to service renegotiation or provisioning.

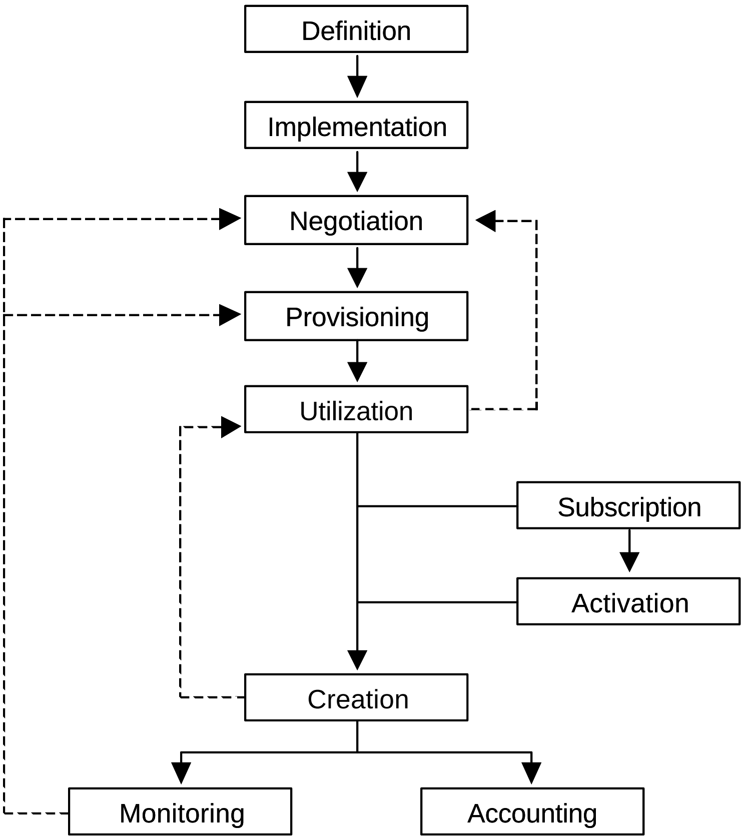


Figure 2.3 – Service life cycle model

The proposal of the Chameleon architecture described in Chapter 3 deals with the first four phases and the last one, namely service definition, implementation, negotiation, provisioning and monitoring. It focus on service definition and negotiation. The other phases (utilization, accounting and monitoring) are important too. They are not considered in an attempt to keep reasonable the scope of this thesis.

2.3 Approaches for QoS in the Internet

Advanced services need to be mapped into IP QoS technologies in the implementation phase, in order to escape from the service-blind nature of the best effort service. In the last years, several approaches for dealing with QoS at the network level have been proposed [184]. This section will focus on the main proposals being currently developed in the context of the IETF (Internet Engineering Task Force) [132][218]: IntServ/RSVP, DiffServ, MPLS, QoS Routing and Traffic Engineering. It also includes over-provisioning as a QoS enforcement technique.

2.3.1 Integrated Services

The integrated services (IntServ) model [31] aims at providing end-to-end QoS guarantees for individual traffic flows by means of reserving resources in every router all the way from source to destination. In addition to the basic best effort service, IntServ adds two new service classes: guaranteed QoS service [182] and controlled load service [217]. The guaranteed service provides firm bounds (mathematically provable) on delay and throughput for intolerant applications (that cannot adapt by any means to the network conditions). It was developed for providing a behaviour similar to leased lines for IP users, but it has not have been deployed due to its high complexity. The controlled load service is a sort of predictive service that provides QoS guarantees similar to those experienced by a flow in a lightly loaded network. The idea behind the IntServ model is that there are no QoS guarantees without resource reservation.

IntServ proposes an implementation framework with four components: the signaling protocol, the admission control routine, the classifier and the packet scheduler. Applications requiring guaranteed service or controlled load service must set up end-to-end paths and reserve resources before transmitting their data. The admission control component is the responsible for deciding whether there are sufficient resources for granting the request. Whenever an IntServ packet arrives at a router, the classifier identifies that packet and puts it on a specific queue to be forwarded. Finally, the packet scheduler will schedule the packet in order to achieve its QoS target.

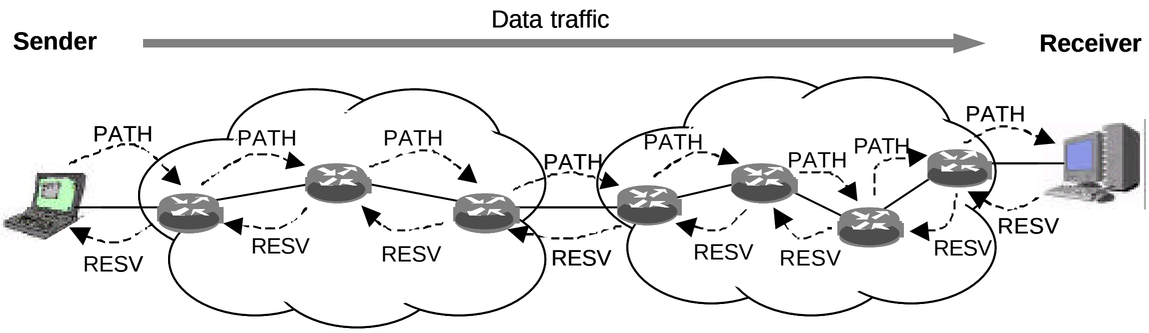


Figure 2.4 – RSVP message processing

Although any signaling protocol may be used with IntServ, RSVP (Resource Reservation Protocol) [30] is the de facto standard. RSVP is a receiver oriented soft-state protocol, developed for applications to reserve resources in one direction in an integrated services network. It is based on the exchange of two messages by sender and receiver: the PATH and RESV messages (Figure 2.4). The sender sends a PATH message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop, determined by the routing protocol. Upon receiving a PATH message, the receiver answers with a RESV message to actually request resources, if it is willing to accept the service request. Every router along the path can accept or reject the RESV message, according to the admission control routine. When a request is rejected, an error message is sent back to the sender. Otherwise, the amount of requested resources is reserved and the RESV message is passed to the next hop. Reserving resource implies installing state information for identifying every flow in their requirements. In order to cope with the routing instability of the Internet, RSVP is a soft-state protocol, that is, the reservations are only valid for a period of time. Therefore, receivers must continuously send new PATH messages for keeping the reservations active.

The IntServ/RSVP architecture represented a major advance in the Internet, which was based on the concept that only end systems should maintain flow-related information. However, this model comes at some cost. Basically, two problems have been identified: 1) It generates too much state information for keeping the reservations active; 2) Too many signaling messages are exchanged for providing flow-level guarantees. In conjunction, these two problems have a negative synergetic effect. It is believed that in the Internet core there are hundreds of thousands of simultaneous flows, which would impose a huge processing and storage burden on the routers. Therefore, although IntServ has not been implemented in the Internet core, it is widely accepted that it is not scalable enough for it.

2.3.2 Differentiated Services

The differentiated services (DiffServ) model was proposed to overcome the scalability limitation of IntServ: no state information and no signaling messages. The essence of DiffServ is dividing traffic into different service classes and giving different treatment to each class. DiffServ aims at being scalable by aggregating a group of flows into a behaviour aggregate (BA), provisioning routers with different resources for these aggregates, and separating the functions of border and core routers. Border router exchange packets with other domains, whereas core routers only have access to internal connections. A domain that implements DiffServ is called a DS domain.

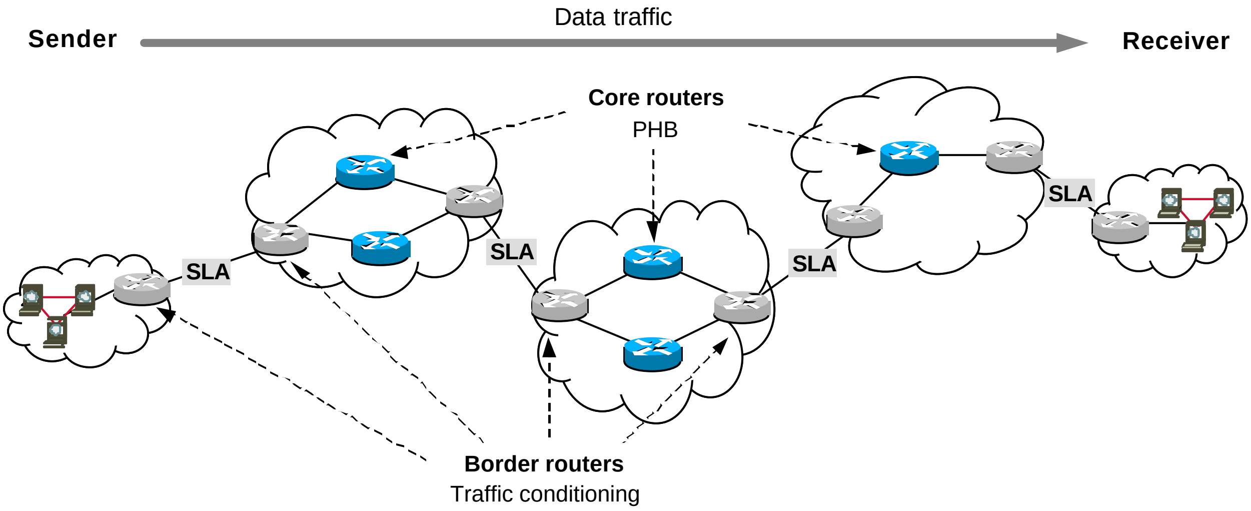


Figure 2.5 – DiffServ architecture

Routers give a specific treatment for packets of different BAs, Known as the Per-Hop Behaviour (PHB). Packets are identified to as belonging to a given PHB through a particular DiffServ Codepoint (DSCP), marked on the DS Field (old TOS and Traffic Class fields in the IPv4 and IPv6 headers, respectively). Border routers perform traffic conditioning and they are allowed to keep per-flow information. Traffic conditioning is needed in order to ensure that the incoming or outgoing traffic is within the bounds of an agreed-upon profile. DiffServ assumes that DS domains had previously negotiated a SLA when they forward traffic to each other. Using the SLA, a traffic profile is taken for configuring the border routers. On the other hand, core routers only examine the DSCP, map it to a PHB and give the packet the adequate forwarding treatment. Border routers also process the PHB. Similarly to IntServ, the guarantees provided by DiffServ are only for one direction, Figure 2.5 depicts the DiffServ architecture.

Formally, a BA is defined as a collection of packets with the same DSCP crossing a link in a given direction in a DS domain. The quantity of packets belonging to a BA may change, as applications send packet bursts to the network. Routers allocate resources for BAs according to the PHBs they are mapped to. The simplest form of implementing a PHB is by allotting to it a percentage of an output link bandwidth.

Two main PHBs were defined for DiffServ: Expedited Forwarding (EF) [59] and Assured Forwarding (AF) [102]. In addition, DiffServ defined one PHB for the best effort service and eight PHBs for compatibility with old implementations [150]. The EF PHB provides strict QoS guarantees for applications sensitive to variations of the temporal network characteristics. It is DiffServ's approach for building a service with low delay, low jitter and guaranteed bandwidth from the network. Obviously, as DiffServ deals with traffic aggregates, even the EF PHB is not able to provide firm guarantees for all flows within an aggregate.

The AF PHB is in reality a group of 12 PHBs. Four AF classes were defined, each one with three levels of drop precedence. AF classes are not associated with levels of guarantees. The treatment received by AF packets depends completely on the link capacity provisioned for it. When an AF class receives a good deal of the bandwidth share, it can even be used for deploying demanding multimedia applications. On the other hand, if an AF class is short of resources, packets can be treated worst than those of the best effort service. Within an AF class, the three levels of drop precedence determine which packets will be discarded first when the network is congested.

As mentioned before, the activities performed by the border routers on every packet for verifying its conformance with the traffic profile (a part of the SLS) are collectively called traffic conditioning. As depicted in Figure 2.6, traffic conditioning involves packet classification, traffic metering and one or more subsequent actions (marking, shaping or dropping).

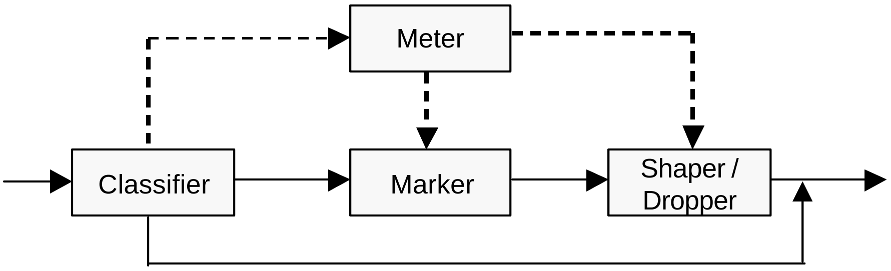


Figure 2.6 – Traffic conditioning

The classifier is the first stage of traffic conditioning. It selects packets received at the input interface based on the content of some part of their headers. If it only examines the DS Field, it is called a BA (Behavior Aggregate) classifier. On the other hand, if it is allowed to analyze other fields, such as IP addresses and ports, it is called a MF (Multi-Field) classifier. The meter is responsible for measuring the temporal properties of a packet flow selected by the classifier according to a specified traffic profile. Among some possible implementations of the meter, the most common is the token bucket [190]. As a result of the metering, packets are sent to a stage where an action is performed, depending on the service. In general, no action is performed on in-profile packets. On the other hand, if the packet is out-of-profile[9], actions including shaping, dropping or marking can be applied. The shaper delays out-of-profile packets until they are in-profile, whereas a dropper discards them. A marker may be used to demote a packet to a low priority PHB.

Intradomain or end-to-end transport services in DiffServ can be configured by combining the effect of traffic conditioning at border routers with PHB enforcement at core routers. Two examples of DiffServ services are given below. A Premium service [151], suitable for multimedia application, can be implemented by using a policer (a meter plus a dropper) and the EF PHB. An Olympic service could offer three classes of differential treatment: gold, silver and bronze. Gold packets receive a better treatment than silver packets that in turn also receive a better treatment than bronze packets. In each class, two levels of drop precedence are defined: high and low drop precedence. The implementation of the Olympic service can use the three classes of the AF PHB, which are allotted a different share of the network bandwidth. At the border routers, the traffic conditioner is based on a classifier, a meter and a marker. Depending on the result of the meter, packets are marked for a particular PHB.

In terms of the service offering model, the difference between IntServ and DiffServ is that IntServ provides and end-to-end service, which provides individual guarantees based on resource reservations for data flows signaled using the RSVP protocol. On the other hand, DiffServ provides a per-hop service for traffic aggregates based on packet prioritization that requires careful provisioning for achieving the intended results.

One of the main issues for dealing with advanced services in DiffServ is how to build end-to-end service based solely on per-hop services and traffic conditioning mechanisms. Since there are problems with the relationship of domains in the Internet, it is likely that advanced

[9] Let us suppose the SLA determines that the traffic bound is 1 Mbps. If a packet arrives at the border router in a moment when the aggregate traffic rate is over 1 Mbps, it is considered out-of-profile.

services will be deployed initially within domains. A Per-Domain Behaviour (PDB) [152] tackles exactly the issue of creating intradomain services. A PDB is the expected treatment that a traffic aggregate will receive from “edge-to-edge” of a DS domain. A particular PHB and the configuration of the traffic conditioning components are associated with each PDB. An end-to-end service can be built later by the concatenation of PDBs of a series of neighbouring domains.

2.3.3 Multiprotocol Label Switching

Multiprotocol[10] Label Switching (MPLS) [172] is an advanced packet forwarding scheme, whereby the next-hop is chosen by means of a fixed length label. It represents a break with the basic hop-by-hop forwarding strategy of the Internet. IP is a connectionless protocol. As such, as a packet travels from source to destination, each router makes an individual forwarding decision, using the IP destination address for looking up the next-hop in routing table (section 2.1.6). MPLS permits the creation of label switched paths (LSP), which give the domain a more effective control on where packets are being forwarded through. When the sequence of routers (called label switched router, or LSR) of a LSP are chosen by the network operator (or by an automatic process), it is called an explicit LSP (E-LSP).

When a packet enters a MPLS network, it is assigned a specific 32-bit header, which contains, among other information, a 20-bit label and a 3-bit experimental use field (EXP, that can be used for assigning classes of services to packets). The MPLS header is either inserted between the layer 2 and layer 3 headers or it is mapped to a specific layer 2 existing header, such as the ATM VCI/VPI field. A variety of different strategies can be used for mapping a packet to a particular label. Packets are grouped into Forwarding Equivalence Classes (FEC), by the combination of some criteria such as a routing table entry, an incoming interface, and a DiffServ PHB. All packets of a FEC are mapped to the same label. From this point on, packets are forwarded through the LSP according to its label.

Figure 2.7 depicts the packet forwarding process in a MPLS domain. Each LSR has a forwarding table consisting of a set of Next-Hop Hop Label Forwarding Entries (NHLFE). Each NHLFE contains a label, the next LSR of the LSP, and an action. The border LSR uses the FEC to NHLFE (FTN) table for mapping a packet to a LSP by means of adding a label and choosing the next LSR. The other LSRs in the LSP use the label for indexing the Incoming Label Map (ILM) table. The NHLFE inform the new label (that has changed) and the next LSR. The last

[10] Although MPLS is called “multiprotocol”, so far only the standardization for the IP protocol is being developed.

LSR of the MPLS domain removes the label and uses the IP routing table for forwarding the packet further. In other words, MPLS creates a virtual circuit over an IP datagram network. The creation of a LSP is done by the distribution of labels to every LSR. Different protocols may be used for this purpose, such as RSVP or LDP (Label Distribution Protocol) [11].

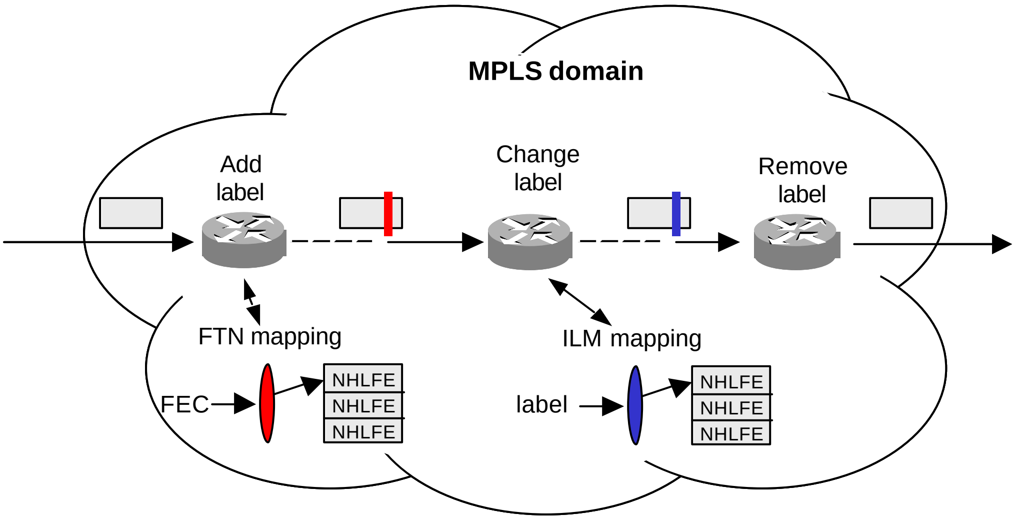


Figure 2.7 – Packet forwarding in a MPLS domain

MPLS is not properly a technique for implementing QoS. Its main application is traffic engineering (2.3.5) [18]. However, when MPLS is combined with DiffServ and Constraint-Based Routing (section 2.3.4), they provide a powerful strategy for QoS provisioning in the Internet [19][219].

2.3.4 QoS Routing and Constraint-Based Routing

Routing in the Internet is focused on connectivity, by finding existing routes for the single best effort service. Current routing protocols, such as OSPF for intradomain and BGP for interdomain, always search the shortest path, usually based on a single metric, most commonly the number of hops. These protocols are "opportunistic" [53], in the sense that they find the shortest path even when it is not the most suitable. Alternative routes are not considered, even though they could be used for routing traffic streams with certain QoS requirements.

QoS Routing (QoSR) [53] is a routing mechanism that selects the path to be followed by a packet flow based on its knowledge of the availability of resources in the network, as well as on the QoS requirement of the flow, such as throughput and delay. Therefore, QoSR can find a longer path, but not overloaded. A QoSR-enabled router has an augmented routing table, with information about the current conditions of the routes, according to QoS criteria. Special QoSR protocols have to be used in order for QoS related information to be exchanged among routers. Some examples are extensions for OSPF [14] and BGP [1][54]. The routing table is extended in two dimensions. First, each entry is extended with the current network conditions according to QoS criteria to the destination address. Second, since there may be more than one path to the same destination with varied QoS conditions, several entries for the same destination may be added.

Constraint-Based Routing (CR) is the process of computing routes that are subject to multiple constraints [218]. One of these constraints may be QoS, so that CR is a concept that encompasses QoSR. CR has evolved from QoSR and sometimes they are used as synonyms. Throughout this thesis, the term QoS Routing will be used for both QoSR and CR. Constraint-Based Routing may be used in MPLS, for setting up paths (CR-LSP) subject to different constraints, including QoS.

QoSR is orthogonal to QoS enforcement and path pinning. QoSR is only able to find paths according to QoS constraints. It does not guarantee that packets will always be routed through a particular path (MPLS), nor it provides QoS guarantees in those paths (IntServ or DiffServ). Therefore, they are complementary approaches.

Most work on QoSR has been done in the intradomain routing context [13]. The reason is that intradomain routing usually considers solely technical aspects. On the other hand, intradomain routing involves different administrative domains, each one using different strategies for engineering routing in their networks. In such an environment, financial and political matters can cause routing inefficiencies, thus impacting adversely the QoS experienced by some portion of the traffic (section 2.1.7). The nature of the interdomain interconnection should first be changed, in order for interdomain QoSR to be effective. The use of interdomain and intradomain QoSR is independent of each other.

Figure 2.8 depicts a scenario where both intradomain and interdomain QoSR are used. Domain A is sending packets to domain D. In domain A, the interdomain routing table of the egress router contains two entries for destination A: through domain B with 1 Mbps or through domain C with 2 Mbps. If the throughput requirement of a given flow in domain A is under 1 Mbps, it can choose both domains B or C as the next hop. If it is between 1 and 2 Mbps, the traffic can be forwarded only through domain C. In case it is over 2 Mbps, no suitable route is found. Domain B participates in the interdomain QoSR, but it does not implement QoSR internally. Therefore, traffic is forwarded by the shortest path. In domain C, intradomain QoSR is being used. The intradomain routing table of the ingress router contains three entries for destination A, with 500 Kbps, 1 Mbps and 2 Mbps. When traffic comes from domain A with a 2 Mbps requirement, then only one path is available.

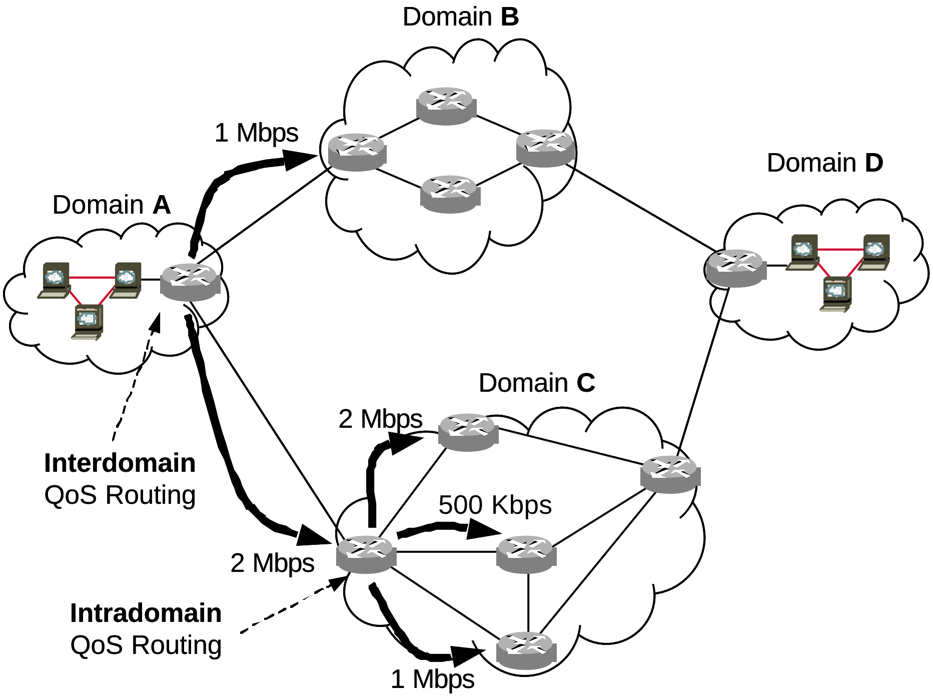


Figure 2.8 – Intradomain and interdomain QoS Routing

2.3.5 Traffic Engineering

Internet Traffic Engineering (TE) deals with the issue of performance evaluation and optimization of operational IP networks [20]. TE can be defined as the process of arranging how the traffic flows through the network, in order to avoid congestions caused by uneven network utilization. A major goal of Internet TE is to provide efficient and reliable network operation while at the same time optimizing network resource utilization and traffic performance. It has become an important design factor in large network projects due to the high cost associated and the competitive nature of Internet and corporate backbones. By carefully changing the normal packet flow through the network, it can be used to achieve the QoS requirements of particular traffic streams.

As discussed in section 2.3.4, the current routing protocols always find the shortest path. The result is that the shortest path for any given pair of sources and destinations usually includes some bottleneck links, due to the network characteristics. Since most traffic is routed through these links, they easily become overloaded. A second negative effect happens when the a flow is routed through a link lacking enough available capacity for giving it a suitable forwarding treatment. These two aspects form the main groups wherein the performance objectives associated with TE can be classified:

- Resource oriented objectives: include the aspects related to the optimization of resource utilization, such as avoiding that some network segments become congested while others have too much spare resources.
- Traffic oriented objectives: include the aspects that enhance the QoS experienced by some traffic streams (e.g., lower delay and higher throughput).

Figure 2.9 illustrates how both resource and traffic oriented objectives can be met by means of engineering paths in the network. There are three different packet flows and three network paths as well. In Figure 2.9a, the network is not traffic engineered and all packets are forwarded through the shortest path. On the other hand, Figure 2.9b shows a scenario where the border router is able to forward packets with different requirements through the most adequate path. Both objectives can be achieved, since the network is more evenly used and the paths are carefully chosen.

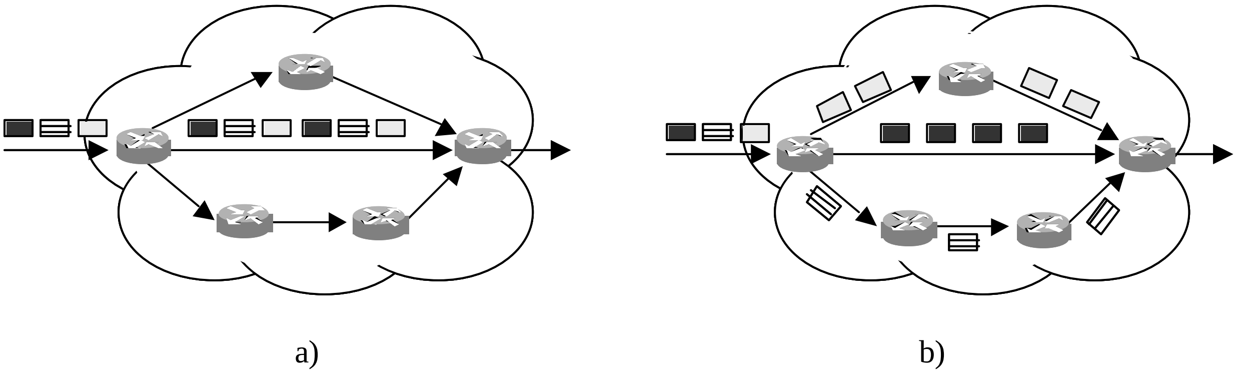


Figure 2.9 – Packet forwarding; a) without TE; b) with TE

Traffic engineering is a process that can be implemented through some different mechanisms, such as: manual route configuration; utilization of particular features of datalink technologies (e.g., by configuring virtual circuits of ATM links); using different weights for the routing protocols; finding out routes with certain QoS requirements using QoSR; and pinning routes with MPLS. Furthermore, TE can be applied to a conventional best effort network, as well as to a network that offers guaranteed QoS service levels, by means of, e.g., IntServ or DiffServ. In a broader sense, TE can be thought of as encompassing all engineering activities related to providing QoS-based advanced services in the Internet and optimizing the network resource utilization.

2.3.6 Over-provisioning

The definition of QoS adopted in this thesis states that QoS is both the performance requirements of the applications and the technologies used for achieving this purpose (section 2.2.2). While the former are the ends, the latter are just the means. The afore-mentioned QoS technologies are based on managing the network resources in order to achieve the ends. However, the adoption of such technologies leads to an unavoidable increase of the complexity of the network. In addition to the higher costs associated with it, a common criticism against managing bandwidth is that the simplicity of the Internet has been the main reason of its scalability and robustness [38][47]. The Internet design follows the end-to-end argument [176], which states that the complexity should be thrown to the end-system, leaving the network itself as simple as possible.

Over-provisioning [4] is an approach for QoS that simply provides enough bandwidth to always match the committed network service guarantees, solely based on the best effort service. The solution consists of throwing more bandwidth on any network link, whenever is seems to be congested. The rationale is that if the network load is kept low, then packets will never be dropped nor they will experience high delay caused by waiting in long queues. Therefore, QoS will be achieved implicitly [55]. As the advances in the last decade in fiber optics and DWDM (Dense Wavelength-Division Multiplexing) made bandwidth widely available and the cost per unit of data transfer fell down, over-provisioning can so far be used continuously, creating the illusion of "infinite bandwidth".

The need for mechanisms for guaranteeing QoS in high speed networks remains a hot debate [218]. Although over-provisioning seems to be very compelling, it is considered a naïve approach by many people, because it considers that every network problem can be fixed with the same remedy: additional capacity. There are various situations where this assumption cannot be proven. A very common belief is that no matter how much capacity there is in the network, sooner or later the users will invent new applications for consuming it. This is commonly referred to as "the tragedy of the commons" [157]. Since there will be no such thing as "infinite bandwidth" ("there is no free lunch"), at some point the network will be congested and the QoS requirements of some applications will not be met, at least for some time periods [118]. Another argument says that no matter how low the cost is, there will be always some cost, and the ISPs will have incentives for maximizing utilization, thus yielding congested links.

2.3.7 Overlay Networks

An overlay network is a “virtual” network created on top of an existing network. The overlay network creates an architecture of a higher level of abstraction, so that it can resolve a variety of problems that are very difficult to be dealt with at the router-level [10]. The nodes of an overlay network are usually responsible for a certain aspect of an entire network or part of it. Examples of areas where overlay networks have been used are multicast, congestion management, peer-to-peer networks, content delivery networks and QoS.

This thesis supports the idea that deploying QoS-based services is not a matter that can be resolved only by implementing mechanisms for resource reservation or service differentiation at the router level. In the last few years, there has been a trend to deal with the QoS problem at a higher abstraction layer, in the form of an overlay network. Some commercial solutions that have been using the overlay technique for offering QoS-based services over the Internet are Internap [111], Virtela [213] and Equinix [68].

A number of research projects have also identified this problem and consequently have focused on building SLA-based overlay networks for QoS provisioning. Most of them consider DiffServ as the basic underlying QoS technology. The Bandwidth Broker (BB) [151] was the first attempt for building an overlay network for resource management in DiffServ networks. Resource provisioning and equipment configuration is one of the most difficult issues involved in deploying a DiffServ network. It can be carried out manually, or semi-automatically. A BB is a logical entity playing two main roles in a DiffServ domain:

1. Intradomain resource management: related to provisioning resources for specific PHBs and configuration of traffic conditioning mechanisms, according to organizational policies.
2. Interdomain resource negotiation: BBs responsible for different DS domains negotiate with each other in order to discover and provision resources for end-to-end advanced services.

The Bandwidth Broker architecture is today’s most accepted solution for resource management in DiffServ networks. It has been adopted by the Internet2/QBone project [192][195].

Three recent projects funded by the European community focus on overlay networks based on SLA/SLS for QoS provisioning [92]. The TEQUILA [198][203] project is meant for deploying services with QoS guarantees and it has been involved with SLS definition, negotiation protocols for SLS and intra-domain and inter-domain traffic engineering schemes so that a network will be able to honor commitments assumed in SLSs. The AQUILA [15][67] project aims to define, evaluate and implement an advanced architecture for QoS in the Internet, which introduces a software layer for distributed, and adaptive resource control. AQUILA is focused on tool construction and trials in real networks with multimedia services. At least at first, service deployment is restricted to just one administrative domain. This means it does not provide solutions for end-to-end service negotiation. The CADENUS [34][52] project proposes an integrated solution for the creation, configuration and provisioning of end-user services with QoS guarantees in IP Premium networks. CADENUS proposes that the QoS provisioning problem be tackled using an overlay network based on mediators: access, service and resource mediators.

SON (Service Overlay Network) [64] adopts a slightly different approach, where the overlay network is in practice a separate provider that buys capacity from underlying transport service domains. The OverQoS architecture [189] follows the same idea.

However, overlay networks also have their pitfalls. They may be subject to a DoS (Denial of Service) attacks, which can lead to a service outage. In addition, network carriers do not appreciate the idea of an overlay network controlling the service operation. They argue that the overlay may not always be able to quickly perform the resource reprovisioning in case of a sudden change in the underlying transit circuit (for instance, in the case of a transoceanic link being replaced by a satellite one). Since these potential drawbacks have not yet been confirmed by experimentation, they do not significantly undermine the deployment of overlay networks for a variety of purposes, including QoS.

2.4 Summary

Although users, providers, vendors and researches want QoS to be deployed in the Internet, it is not a reality today. The obvious conclusion is that something is still missing, in spite of the progress with proposing new technologies and incorporating them into commercial products in the last years. The goal of this chapter was to show that some of the missing pieces of the QoS puzzle are:

- The development of new solutions for the current problems with the interconnection of domains in the Internet. Particularly, business relationships and settlement structures are needed, in order to provide domains with the right financial incentives for the QoS deployment.

- A comprehensive understanding of the life cycle of advanced services in the Internet, having in mind that any successful solution has to be driven by sound business models and not by the enabling technologies. The negotiation of contracts (SLA/SLS) should be at the heart of any proposal involving end-to-end services.

- Different domains can opt for different QoS technologies or a careful configured combination of them for transcending the limitations of the best effort service. A solution should explicitly allow domains to choose an adequate technology, as long as they can meet the negotiated features of the high level abstract services.

The next chapter presents the Chameleon Architecture, which aims at adding some contributions to the deployment of QoS in the Internet, based on the afore-mentioned findings of this chapter.

Chapter 3

The Chameleon Architecture

In this chapter, an architecture for enabling the deployment of QoS-based advances end-to-end services in the Internet is proposed and it is called the Chameleon[11] architecture. As stated in Chapter 2, obtaining end-to-end QoS is not an easy task, due to limitations of the technology, topology and financial settlements. Chameleon is an overlay network, which means that it tries to resolve some of these problems by means of higher level entities in charge of a series of activities related to the service life cycle, proposed in section 2.2.6. The proposal of the Chameleon architecture deals with the phases of service definition, implementation, negotiation, provisioning and monitoring. Most work in this thesis was devoted to the phases of service definition and negotiation, described in more details in Chapter 4 and Chapter 5, respectively. The high level design of the Chameleon architecture presented in this chapter is one of the main contributions of this thesis.

In the sequence of this chapter, section 3.1 presents an overview of the Chameleon architecture, which puts the above-mentioned service phases into three logical planes. Sections 3.2, 3.3 and 3.4 present a more in-depth view of each plane, namely the service, operation and monitoring planes. The resource control functions of Chameleon are discussed in section 3.5. Section 3.6 analyses the question of whether an overlay architecture such as Chameleon will be necessary in the context of network over-provisioning. The overview of an example scenario of a simple implementation of the Chameleon architecture is presented in section 3.7. Finally, section 3.8 draws some comments on the main topics discussed in this chapter.

[11] The architecture is called Chameleon due to its goal of adapting to any existing underlying QoS technology, as long as it is able to maintain the semantics of the end-to-end services, as there are seen by the users.

3.1 Overview of the Chameleon Architecture

The Chameleon Architecture is an overlay network aimed at providing QoS-based advanced end-to-end services in the Internet (Figure 3.1). It is divided up into three logical planes, in order to offer flexibility to service definition and negotiation, efficient implementation and control of proper operation of the contracted services:

- Service Plane: creates an abstract model of the network, so that domain may have similar external interfaces for service definition and negotiation.
- Operation Plane: through this plane domains implement the negotiated services by means of a given QoS technology and any other control and data path mechanisms.
- Monitoring Plane: it is orthogonal to the other two planes and its main responsibility is ensuring that services are operating within the agreed upon quality thresholds.

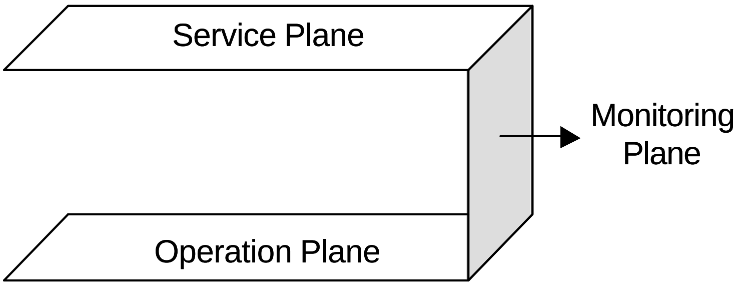


Figure 3.1 – The Chameleon architecture – conceptual view

This organization in three planes provides a homogeneous and integrated view of the network, although the actual deployment of an end-to-end service may need the cooperation of many networks that possibly implement distinct QoS technologies. This design is inspired on the dual view of QoS as being both the performance guarantees for applications and the underlying technologies for enabling the network to provide such guarantees (section 2.2.2).

However, the actual motivation for a three-plane architecture comes from Chameleon's main objective, which is building an overlay network for deploying QoS-based advanced services in the Internet. The operation plane is comprised of almost everything that is used in the current Internet (i.e., protocols, mechanisms, routers, etc.). Since so far the Internet did not agree on a way for deploying new advanced services, this work argues that careful treatment for the higher level aspects of service deployment is required, instead of excessively focusing on router matters at the underlying network. Therefore, Chameleon was designed with the strong concern in clearly separating the operation and service planes. The reason for a third plane, targeted to handle monitoring tasks, is more a conceptual division than a specific need. It is not essential, though very important for assuring the quality of the deployed services. In practice, monitoring activities are closely related to the operation plane. In any case, the Chameleon architecture can be summarized in only one entity, the Service Manager (section 3.1.1), which can be implemented as a single software entity.

The Chameleon architecture may be seen as an overlay network that extends the Internet architecture in order to deploy advanced services. It follows the model of network architectures based on layers (i.e., planes), interfaces and protocols [190], even though informally. To a large extent, the description of the architecture is dedicated to explain these aspects, their relationships and the internal components. In addition, it presents some suggestions for prospective future implementations.

A network that participates autonomously in the deployment of Chameleon is called an administrative domain, or only domain for short. A group of domains willing to deploy end-to-end advanced services together (a Chameleon Domain Group, or CDG – section 3.1.2) necessarily need to agree on adopting some standardized services, called Well-Defined Services (WDS - Chapter 4). Periodically, domains need to renegotiate the offered services in an automatic way, in order to be able to adapt to dynamic scenarios and constant variations of traffic volumes (Chapter 5). During the negotiation process, domains can play the role of both service buyers and sellers.

3.1.1 The Service Manager

The Service Manager (SM) is an entity that implements the required functions for making it possible to deploy and use the Chameleon architecture. The SM has the responsibility over all computational activities that happen in the service, operation and monitoring planes. To this end, it is internally divided up into three components (Figure 3.2):

- Service Broker (SB): it is the responsible for implementing the mechanisms of the service plane, such as service negotiation and high-level resource management (section 3.2.2).

- Resource Manager (RM): it is the responsible for implementing the functions of the operation plane, including resource reservation, buffer management and traffic conditioning (section 0).

- Monitoring Coordinator (MC): it is the responsible for implementing the functions of the monitoring plane, such as traffic measurement, statistical analysis and alarm generation (section 3.4.2).

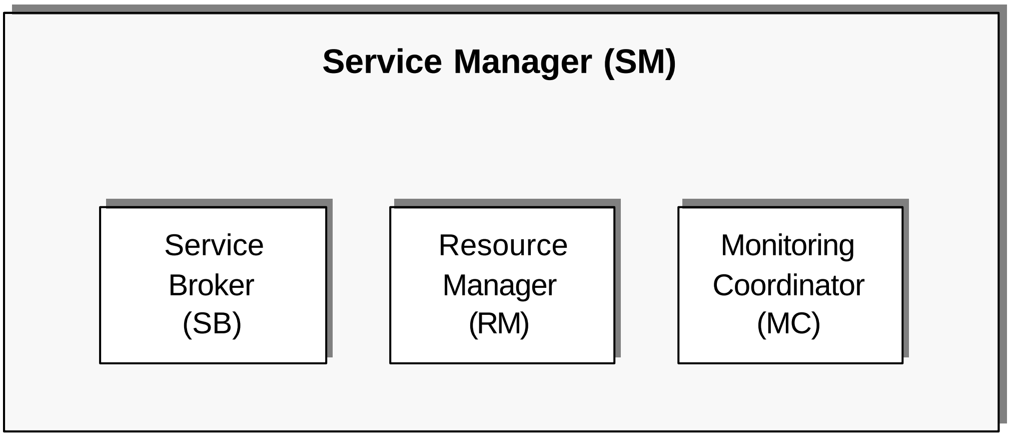


Figure 3.2 – The Service Manager

The SM is a logical entity, in other words its implementation can be distributed to various hardware and software components in order to obtain better performance, scalability, flexibility and robustness. Since it is a single point of failure, the SM can easily cause problems to a domain, in case it goes down. Obviously, the implementation decision depends on the particular dimensions and necessities of each domain, as it may imply additional complexity and costs. One that satisfies the above-mentioned requirements is by using a hierarchical DNS-like structure with distributed and replicated servers. Hierarchical implementations for resource management in IP networks have already been proposed in the literature [163][221].

An implementation also can combine strongly related functions of distinct planes in the same entity, at the discretion of every domain. As an example, the service plane function of resource management and the operation plane function of network dimensioning are related, although they are conceptually in separate abstraction levels. For the same reason, some functions of the monitoring plane can be implemented together with the functions of the service and operation planes.

3.1.2 Chameleon Domain Group

A group of domains willing to implement the Chameleon architecture in order to jointly deploy advanced QoS-based services is called a Chameleon Domain Group (CDG), and every domain associated to it is called a CDG member. The concept of CDG is intentionally loose and abstract. It only exists to formalize the idea that an important step for deploying QoS in the Internet is getting the separate domains cooperating with each other. If there is no true collaboration, at least there must be commercial and technical multilateral agreements that are deemed acceptable by all CDG members. Even in a highly competitive environment, domains need to agree on some points in order to deploy advanced services; in much the same way they currently have connectivity agreements. All domains in a CDG must sign a multi-lateral SLA (section 5.3.1).

The duties, responsibilities and rights of the members and the relationship among them may vary substantially from one CDG to another. It depends on the decisions taken by each CDG, such as the choice of the service negotiation model, which can have a strong influence on the level of interdependence of domains. A CDG is not a NAP, in other words being a CDG member does not mean than a given domain peers with every other CDG member. Instead, CDG members may rely on the same interconnection structure they had before the CDG was formed. The intention here is exactly to interconnect domains that do not necessarily have a direct peering interconnection.

There is no rule for the creation of new CDGs. A group of domains may themselves decide to form a CDG in a more collaborative way or a business-oriented entity may do it for them. Domains can be grouped on a geographical scope basis (city, state, region, country), commercial interests or any other criterion. A domain may take the decision for getting associated to a particular CDG based on some potential, such as the set of deployed services and existing CDG members. There is no limit imposed on the number of CDG members and CDGs a domain can be associated to. A CDG may be comprised of as little as two domains, or it could encompass the whole Internet (at least in theory).

Every CDG needs a CDG Controller responsible for its operation, independently the number of members and the nature of its activities. The CDG Controller can be a CDG member or an outsourced organization. The main responsibility of a CDG Controller is maintaining a repository of information needed for the CDG operation. This repository should contain the registry of CDG members, the standard SLA for the multi-lateral agreements, the current negotiation model used by the CDG, the set of services deployed by the CDG, the subset of services implemented by each CDG member and the identification of each member. Domains must declare their “basic” network address to the CDG in the form of *address/prefix-length.* Their traffic will be identified according to this address. Optionally, domains may declare network addresses of internal clusters or Points of Presence (PoP). The cluster addresses may or may not belong to the range of the basic network address. In order for a packet to be identified as belonging to a domain, its (source or destination) address must be in the range of either the domain or one of the clusters.

Some other responsibilities that can be delegated to the CDG are: a) establishing and negotiating agreements with other CDGs for expanding its scope (section 5.10); b) controlling the quality of the delivered services by maintaining a database with information gathered from the monitoring plane of each domain; c) performing the settlements among domains, while acting as a clearing-house.

3.1.3 Scope of End-to-End Services

The Chameleon architecture provides end-to-end services, which scope extends from host to host, regardless of the number of intermediate domains the traffic can cross. From a user point of view, the service is actually the user service, and he/she should not be aware of the existence of transport services. Figure 3.3 illustrates the user's view of an end-to-end service, through the combination of a transport service provided by domains using different QoS technologies in the operation plane.

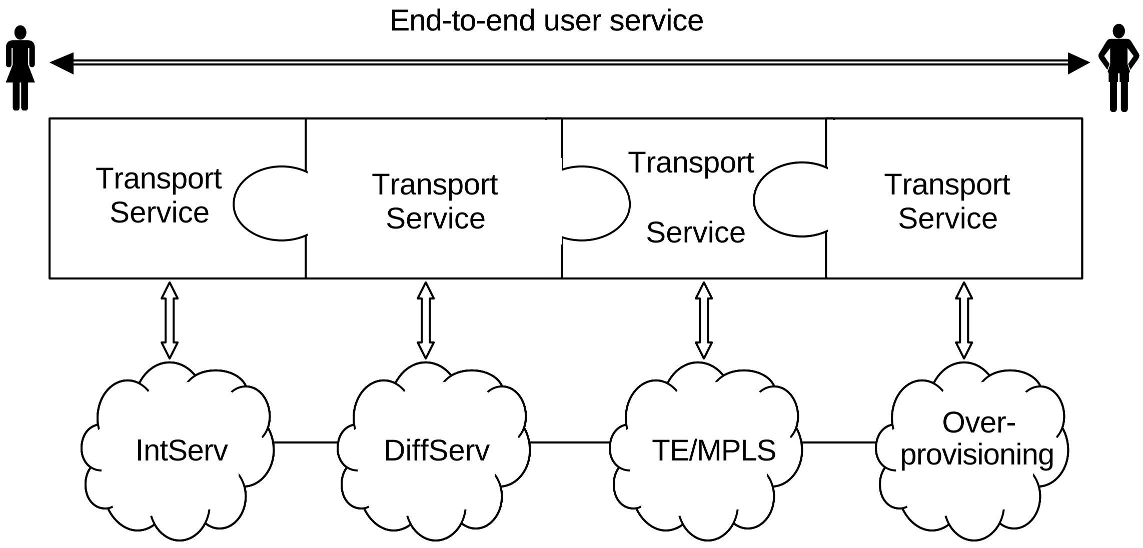
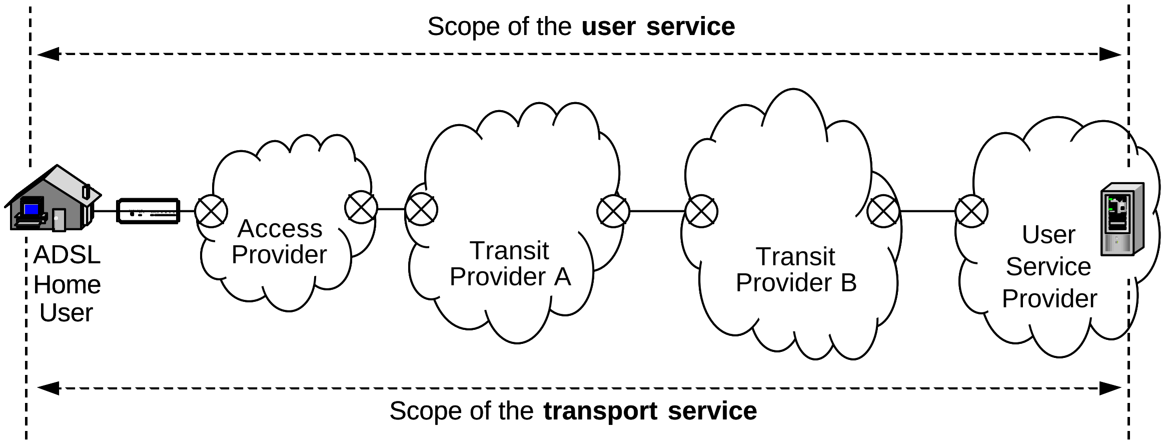


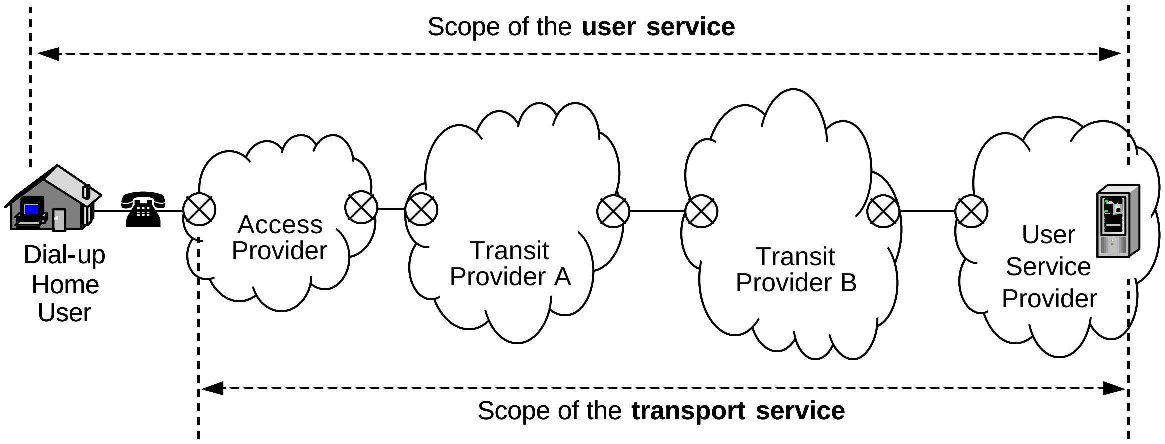
Figure 3.3 – User's view of an end-to-end service

Chameleon is only able to provide QoS guarantees for transport services, limited to a CDG area. Extending guarantees to the end-user is a responsibility of each domain. However, it can be difficult in some conditions, mainly for some access networks or customer premises. For example, an access provider may not extend the guarantees for some interactive multimedia services to dial-up users, due to limitations such as low bandwidth and high delay. Similarly, an access provider may not be able to extend guarantees to a corporate network. It is up to the customers to manage their own network in order to provide the adequate QoS guarantees, unless the network management is outsourced to the access provider.

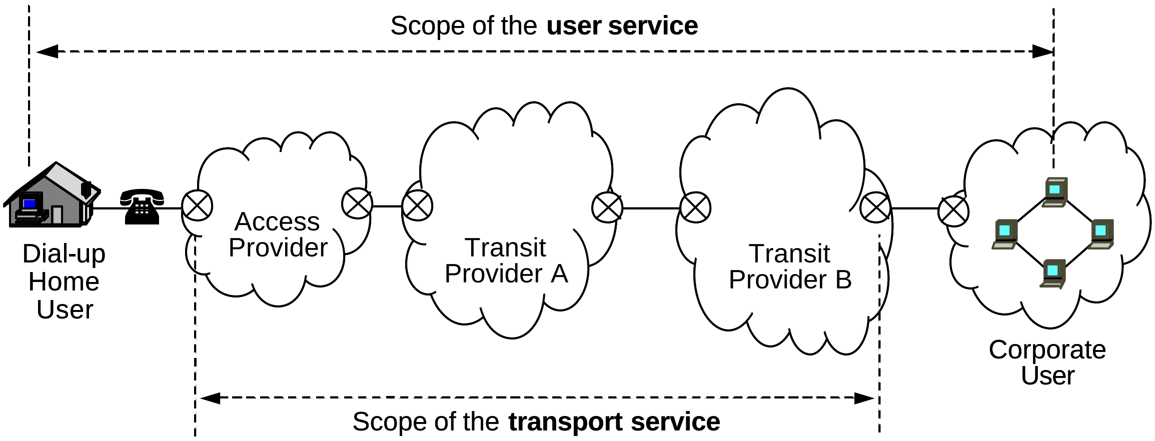
Figure 3.4 shows three scenarios for the scope of user and transport end-to-end services. In Figure 3.4a, the User Service Provider implements Chameleon and is a member of the CDG, and the user access network is based on ADSL. Therefore, the guarantees of user and transport services are maintained, extending from the USP server up to the user home PC. Figure 3.4b shows a slightly different scenario, whereby the user is connected through a dial-up line, and the guarantees for the transport service only extend to the access router. In Figure 3.4c, both end-users are not covered by the transport service guarantees. Therefore, the scope of the transport service extends to the access routers from both access providers.



a)



b)



c)

Figure 3.4 – Scope of user and transport end-to-end services

3.1.4 Chameleon Interfaces

In the Chameleon architecture, the communication among planes is done through interfaces with well-defined functions. The usual notion of interface of network architectures is extended to a broader concept that, for instance, allows the direct communication between each pair of planes. Chameleon interfaces can be grouped into two types, internal and external interfaces, as depicted in Figure 3.5.

External interfaces refer to interdomain communications and turn out to be in the same plane of separate domains. The common understanding in network architectures is that external interfaces are comprised of protocols. In Chameleon, this concept is broader, also encompassing service negotiation models (section 5.4) and techniques for service monitoring (section 3.4). Three types of external interfaces have been defined (Figure 3.5):

- S-S (Service- Service) interface: interface among service planes of distinct domains, involving service definitions, negotiation models, negotiation protocols and algorithms. Elements of the S-S interface are presented in more details in section 3.2, Chapter 4 and Chapter 5.
- O-O (Operation- Operation) interface: it is comprised of the protocols currently used in the Internet in all levels for domain interconnection, that in Chameleon are in the operation plane. Some examples are protocols for data exchange, (IP, TCP, UDP, SCTP, etc.), routing (BGP) and control (ICMP, RCTP, etc.). The intention is not to define new operation protocols, but to use existing ones.

- M-M (Monitoring- Monitoring) interface: this interface implements the functionalities of the interdomain monitoring (section 3.4.6), which involves defining new protocols, metrics (section 3.4.3), measurement points (section 3.4.4) and measurement frequency (section 3.4.9), among other aspects.

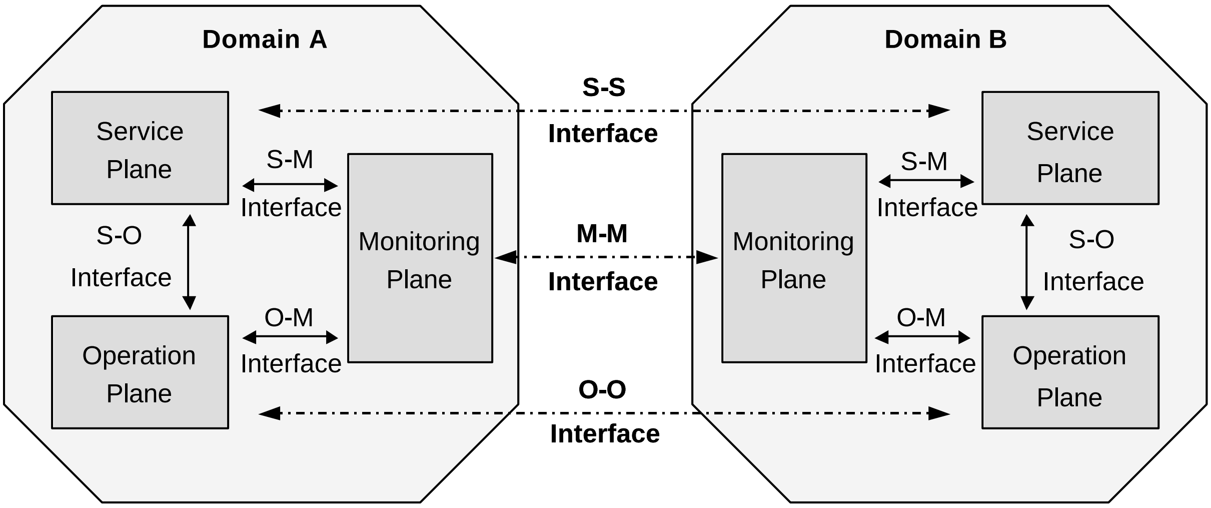


Figure 3.5 – Internal and external interfaces of Chameleon

Domains of a CDG must use the same external interfaces, although they do not necessarily need to be standardized (by the IETF, for instance, which offers a number of standards for the operation plane). Since Chameleon assumes that domains use different QoS technologies in the operation plane, simple and well-defined interfaces are essential in order for the end-to-end services to present the expected results and guarantees.

Internal interfaces always involve two distinct planes of a given domain and basically consist of APIs. Defining internal interfaces is a useful guide for making it possible to distinguish the limits and relationships among the planes, and the functionalities offered by one plane to the others. In a given interface, the functions can be invocated by both involved planes. In other words, planes are not hierarchical as with the layers of network architectures, where one layer always offers services to another higher-level layer. Similarly, to the external interfaces, there are also three types of internal interfaces (Figure 3.5):

- S-O (Service-Operation) interface: interface whereby the service and operation planes offer services to each other. The functions belonging to the S-O API refer to either commands or queries from the service plane to the operation plane for resource provisioning, or to queries from the operation plane to the service plane in order to get enough information for capacity planning (section 3.3.1).

- S-M (Service-Monitoring) interface: defines the relationship between the service and monitoring planes. The service plane uses the S-M API mainly for getting traffic samples for traffic prediction. Conversely, the monitoring plane uses it mainly for sending alarms and performance reports.
- O-M (Operation-Monitoring) interface: this interface involves the information exchange between the network dimensioning module of the operation plane, responsible for physical resource provisioning, and the alarm generation and performance reports modules of the monitoring plane.

Standardization of internal interfaces is not needed, because they are merely related activities that happen inside domains. Nonetheless, some formal "guidelines" may be useful, since they can be applied in various domains, making the implementation easier.

The implementation of a plane can be completely transparent to the other planes inside a domain, with every information exchange among planes happening through the internal interfaces. In practice, an implementation can be considerably more complex and inefficient if the strict separation of planes is followed. This is the case of the functions of the monitoring plane, which are closely related to the service and operation planes. As such, domains are free to combine functions of distinct planes for getting a more simple and efficient implementation.

A similar rationale is valid for the relationships among domains. The external interfaces (mainly the S-S interface) also seek to hide information about the internal structure of each domain, such as topology, link capacity and technology. However, domains can violate this encapsulation and hence lose its inherent its benefits at their discretion. Although this is not an ideal situation, this kind of flexibility is important to the success of the Chameleon architecture, since its main goal is to provide a simple framework for the evolution of the current Internet into a network capable of offering advanced QoS-based services.

3.2 The Service Plane

The service plane plays a fundamental role in the Chameleon architecture, mainly because it is involved with the management of advanced services (service definition, service negotiation and resource provisioning), in order to support the abstract view for obtaining a seamless and homogeneous service when combining multiple domains in an end-to-end scope. This plane implements the external interface of the Chameleon architecture, called the S-S interface. The aspects of the service plane that should be standardized within the Chameleon architecture are limited to the S-S interface.

The service plane creates an overlay network over the operation plane, which is basically represented by the current Internet infrastructure, made of routers, hosts, links and other network components. The main outcome of this feature is that the service plane can be implemented in any domain without changes to the existing network, except for the introduction of QoS technologies used for providing performance guarantees to advanced services.

3.2.1 Functions of the Service Plane

Most activities of the service plane are purely computational, but some of them also require involvement of humans (the service administrator), while performing activities such as business strategies, definition of policies, rules and constraints for the interdomain relationship, and definition of internal priorities and policies for the service and network management. The responsibilities of the service plane are distributed over the following functional modules (Figure 3.6):

- Service definition (Chapter 4): Transport services in Chameleon have a formal definition trough Well-Defined Services (WDS). Each domain chooses the WDSs it wants to offer, from those deployed by the CDG. The service plane has the responsibility of maintaining up-to-date information about the offered (user and transport) services, WDSs negotiated with other domains, user services sold to end-users and policies for mapping these services to transport ones. This information is stored in the service repository.

- Service negotiation (Chapter 5): The dynamic and automatic service negotiation among domains represent an essential aspect for the deployment of advanced services in the Internet. The negotiation outcome is stored in the service repository. The interaction with the operation plane responsible for resource provisioning, which occurs after a negotiation, is carried out by the resource control module.

- Resource control (section 3.5): Consists of the activities of resource estimation, service offering and resource provisioning. Resource estimation is mostly based on traffic prediction and is strongly dependent on service purchase. Service offering is based on the resource availability and the state of the operation plane and is related to the service sale. Resource provisioning refers to the amount of resources to be allocated by the operation plane, according to the implementation strategies of each offered service. It is performed as a result of the service negotiation, which is in turn triggered by both the service sale and purchase. The resource control module is constantly in contact with the network dimensioning module of the operation plane (through the S-O interface) in order to obtain information for the service offering and to request the resource provisioning. It also communicates with the capacity planning module, providing information about future resource demands.

- Traffic prediction (section 3.5.4): The ability to predict future traffic demands is essential for the appropriate resource provisioning and for the deployment of advanced services. Traffic prediction provides some interesting features to the resource provisioning, namely stability, robustness and scalability.

- Admission control (section 3.5.5): Some user services, such as interactive voice and video, need admission control at the time they are invocated. Different admission control schemes may be used, depending on the type of service [121][135][162]. In case of measurement-based admission control (MBAC), the information of the current network load is obtained from the load evaluation module of the monitoring plane, whereas in case of parameter-based admission control, the information is obtained from the service repository.

- Traffic sampling: The monitoring plane collects traffic samples from the domain border routers, through a simple protocol, such as SNMP. The service plane requests statistics on these samples (for instance, average and standard deviation) from the statistics module of the monitoring plane and stores them in the traffic repository for future processing.

- Historical indexes: This is an off-line activity that consists of analyzing the traffic statistics stored in the service repository and generating historical indexes of hourly, daily and monthly demand that affect the traffic estimation. Traffic volume variation over different timescales is a well-known characteristic in the Internet. This module builds a matrix of indexes that will be used as a multiplier for helping the resource estimation function (section 3.5.1).

- Service/operation mapping: This mapping is needed in order to implement in the operation plane, the services defined and negotiated in the service plane. Domains can use any technique or approach to the service/operation mapping, since it refers to the implementation strategy of the service plane and its relationship with the operation plane. The range and performance of possible solutions can vary from simple static configurations up to the utilization of a highly flexible (and complex) set of policies and their automatic translations into calls to the S-O API.

- Corrective reaction: This function consists of the evaluation of information feedback proceeding from the alarm generation and performance reports modules of the monitoring plane and the coordination of corrective reactions as a consequence of transitory trouble in service operation. The actions taken by this module can be either in the form of immediate reconfigurations in the operation plane (through the resource control module), or yield future changes in the mechanisms for resource provisioning, service negotiation or service/operation mapping.

- Domain discovery: This module has the responsibility for identifying neighbouring domains and establishing communication with them through the S-S interface. This activity can be triggered by a manual configuration from the service administrator (a human) or through some protocol for discovering Service Brokers (section 3.2.2).

- Routing control: This module is involved with getting and distributing routing information, in case service negotiation is able to take routing decisions (section 5.4.6). The service plane interacts with routing protocols, both interior (IGP, such as OSPF and IS-IS) and exterior (EGP, such as BGP).

The full set of functional modules of the service plane need not be present in every instance of the Chameleon architecture. Rather, it is expected that most implementations will only use a sub-set of these functions, which encompass the core functionalities of the service plane. They include service definition, negotiation and provisioning, according to the service life cycle proposed in section 2.2.6. An example scenario for an implementation of the service plane is presented in section 3.7.2.

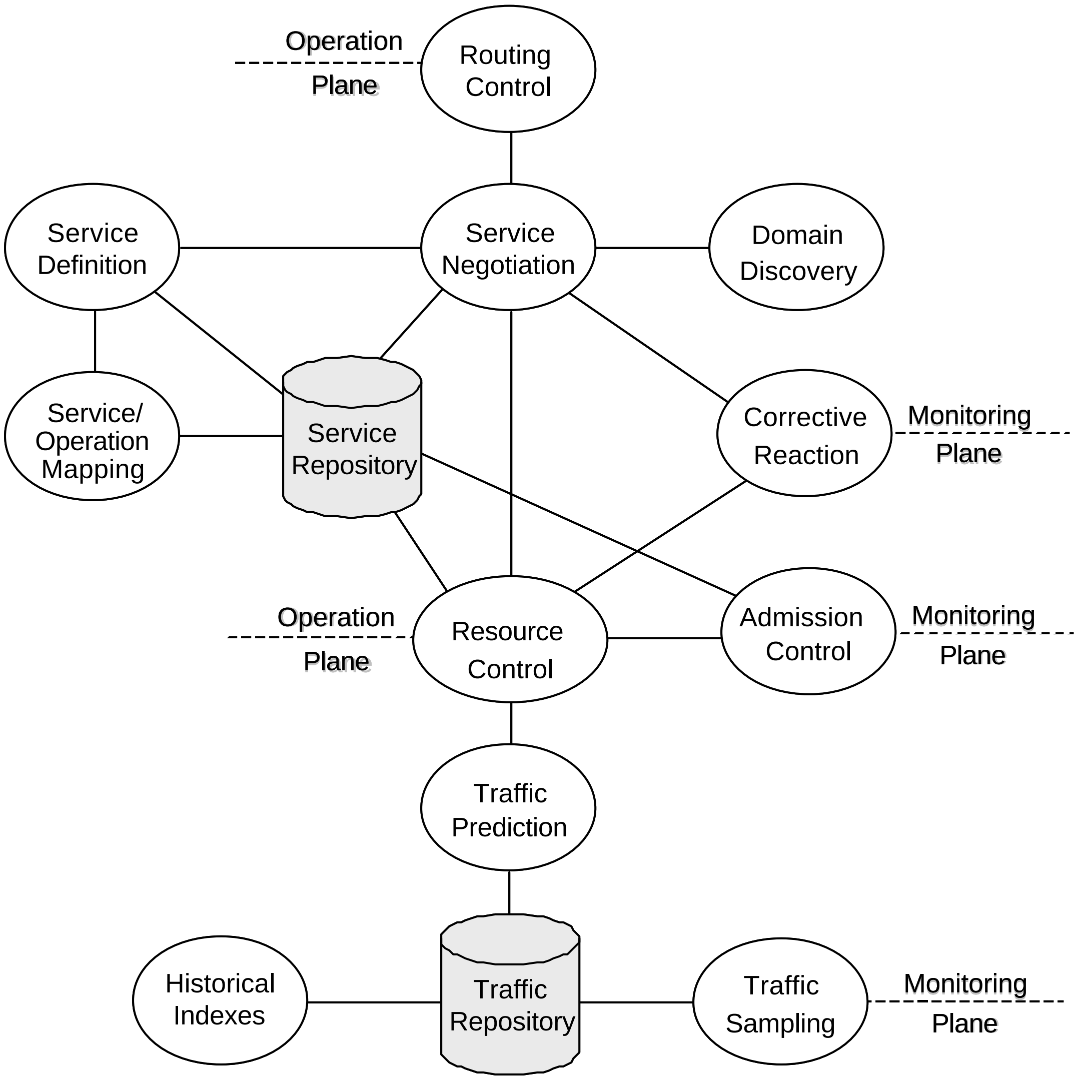


Figure 3.6 – Functional modules of the Service Plane

3.2.2 The Service Broker

The Service Broker (SB) is the entity in charge of every computational activity that happens in the service plane of the Chameleon architecture (the SB is part of the Service Manager, section 3.1.1). The SB contains the implementation of the afore-mentioned functional modules. It can be seen as an extension of the Bandwidth Broker (BB, section 2.3.7), but with three distinctive differences:

1. Unlike the BB, which is focused in DiffServ, the SB is not tied to any particular QoS technology in the operation plane. The SB assumes that the domain is able to implement and fulfill certain negotiated QoS performance guarantees.

2. The SB is able to negotiate services defined by WDSs, that are based on any meaningful combination of QoS parameters (delay, jitter, packet loss and throughput) and not only to perform capacity allocation (i.e., throughput). Whereas the BB is just a resource negotiator for DiffServ-based services, the SB actively participates in both phases of the transport service negotiation process, namely WDS negotiation (section 5.3.2) and resource negotiation (section 5.3.3).
3. The SB is responsible for a variety of other functions of the service plane, as stated earlier. The functions typically performed by a BB, interdomain resource negotiation and internal resource provisioning can be seen as a subset of the functions allocated to the SB.

The SB also maintains two data repositories in the service plane: the service and traffic repositories. In the service repository is stored:

- The set of user services currently offered by the domain (if any) and the mapping policies from user into transport services. For instance, a Telephony user service may be mapped into a Premium transport one. Chameleon is not directly concerned with the choice of user services, nor is it involved in their standardization. Consequently, the type of information related to user services is outside of the scope of this architecture.
- The set of transport services (WDSs) that are being implemented by the domain. To each service, the service repository keeps an identifier (WDSID) and some negotiation parameters (section 4.3.2), such as the maximum amount of resources that can be sold for that service over every link and a minimum percentage of resources that can be granted for that service during the negotiation with other domains (section 5.1.3).
- Information about the service/operation mapping, used for resource provisioning and equipment configuration, according to the adopted QoS technology in the service plane. For instance, the afore-mentioned Premium transport service may be mapped to the EF PHB in a DiffServ network.
- Information about service purchase, including the resource estimation matrix (section 3.5.1), that is requested from other domains through negotiations, the service grant matrix (a negotiation result), reasons for service denial (section 5.1.3), and historical data (previous negotiation results).

- Information about service sale, including the service offering matrix (section 3.5.2), the resulting sold service matrix and also historical data.
- Policies for identifying packets belonging to WDSs in the domain ingress router, mapping the WDS to a local QoS technology (such as DiffServ PHB, MPLS LSP and IntServ class of service) and possibly remapping in the egress router. An example with some practical scenarios is described in section 4.5.
- Information related to performance optimization yielded from local negotiations between a pair of neighbouring domains (section 5.3.1), such as some particular QoS signaling protocol used in their interconnection. For example, an access provider can use a variation of IntServ/RSVP for reserving resources for individual flows up to the ingress router of its transit domain, where they are then mapped to DiffServ PHBs.
- Information needed for admission control: for each service, this may involve the maximum (predefined) amount of resources, the current active (admitted) user sessions, the current available amount of resources, authorization, authentication and accounting information. The maximum amount of resources is determined by service negotiation in the case of parameter-based admission control and by the current network load when using measurement-based admission control. Furthermore, the service repository maintains information about blocked sessions, which is useful for the resource estimation.

The traffic repository is much simpler, but it stores a higher amount of information. It contains traffic samples obtained from the monitoring plane, the historical indexes of traffic demand and the matrix yielded by the traffic prediction module (also including historical data).

3.3 The Operation Plane

The operation plane implements the engineering decisions with respect to resource provisioning and equipment configuration, responsible for making it possible the utilization of the offered services (to internal users and other domains). Provisioning refers to the determination and allocation of resources at various points of the network. Configuration is the distribution of operation parameters to the network equipments in order to achieve the provisioning objectives. The operation plane subsumes functions that normally are assigned to the data and control planes. On the other hand, functions that generally are assigned to the management plane are found in the service plane in Chameleon[12].

Each domain, through an internal set of policies, maps the offered services previously negotiated with other domains to some mechanism or technique that can be used to resource provisioning and equipment configuration. Domains can implement known approaches for providing QoS, such as IntServ, DiffServ, MPLS and over-provisioning. They are also free to configure their networks with any other mechanism or technology, as long as it is able to fulfill the requirements of established contracts for services deployment. In theory, any domain could choose any enabling technology. In practice, however, some constraints should be respected. For instance, IntServ should be used only in stub networks, because of its well-known scalability limitations.

To put it more simply, the architecture makes the service offering independent of the technological decisions. This feature also potentially allows a domain to change its underlying QoS technology without adversely impacting its service offering. This is particularly true if the interfaces among planes are not changed and the new implementation respect the required service guarantees. Thus, the operation plane can be seen as being encapsulated within every domain, and logically separated from the service and monitoring planes.

In the operation plane are protocols and technologies used for exchanging information in the Internet, such as the TCP/IP protocol suite and sub-IP[13] technologies including Ethernet, ATM and Frame Relay. Basically, all protocols and technologies that presently are used in the Internet for carrying data fall within the operation plane in Chameleon. The Chameleon architecture is not intended to develop new technologies and protocols for the operation plane, but makes intensive use of well-known ideas and new developments of resulting from recent research projects based on overlay networks in this area (section 2.3.7 presents some examples).

3.3.1 Functions of the Operation Plane

Most aspects related to the operation plane are covered by Chapter 2. In general terms, the operation plane has two functions (Figure 3.7):

- Network planning: It is concerned with making resources available for satisfying demands of end-users. Some resources must be statically provisioned, for instance

[12] The terms “data plane”, “control plane” and “management plane” are being used here, although this terminology is not a consensus.

routers an interconnection links, which must be procured, installed and configured before they are ready for utilization. When supported by the underlying network, some resources can be dynamically brought from stand-bye to the operational state. This feature has been called network engineering [42].

- Traffic management: It involves the management of the traffic and optionally the network paths where it should be forwarded through, in order to obtain better service performance and network resources usage. This function is also known as Traffic Engineering and depending on the technologies that a given domain adopts, some modules may be more or less important. Here the object is not to promote a detailed discussion about the implementation of the traffic management module in specific technologies, but only to present a general view of this subject.

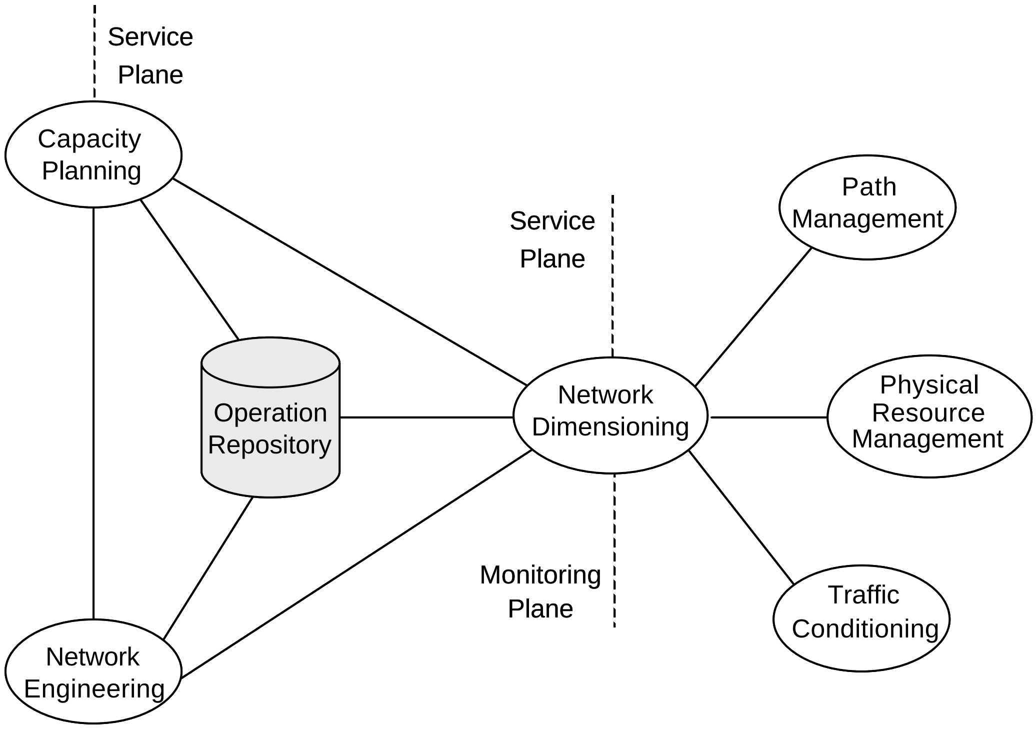


Figure 3.7 – Functional modules of the Operation Plane

The modules that implement the functions of network planning and traffic management are described below. As stated earlier, the idea is to use existing solutions whenever possible; particularly those developed by highly relevant research projects, such as TEQUILA and AQUILA (section 2.3.7):

[13] Any technology used for conveying IP traffic is called sub-IP.

- Capacity planning: This module refers to the long-term planning of the physical network resources. It implies the procurement and installation of new equipment, such as routers, switches and servers, and also the acquisition or upgrade of interconnection links (and even the extension or improvement of the cabling system, in the case of a LAN). New resources must be added to the network in order to prevent or fix situations where the network is persistently overloaded (in a timescale of weeks or months), causing congestions that cannot be solved with known techniques of resource management [134].

- Network engineering: It is an automation of part of the IP network planning, generally executed off-line, when this feature is supported by the underlying network [42]. Therefore, it is not a mandatory module of the Chameleon architecture. It allows making changes in the network topology in real time, within certain bounds, by dynamically configuring optic fiber links according to current and predicted traffic demands.

- Network dimensioning: This module is responsible for the function of mapping traffic to network physical resources and also configuring the network to accommodate the traffic demand predicted for the short-term future (timescale of a few hours or days). Its main task is to compute the amount or resources needed in routers (buffer space) and links (bandwidth share) for allowing the traffic to be forwarded without violating the required QoS guarantees. In order to be effective, the network dimensioning module needs [169]: a) traffic models, which define the traffic generated by the users in statistical terms; b) traffic description parameters, also known as traffic matrix, which describe the expected traffic load between each internal network path (a pair of ingress and egress routers); c) dimensioning algorithm, which uses the traffic models and the traffic matrix to compute the required amount of resources in routers and links. Actually, the traffic matrix is a resource provisioning matrix, which contains the resources that should be provisioned as a result of the service negotiation. The network dimensioning module has a strong interaction with the resource management module of the service plane, through the S-O interface. It executes commands for the resource provisioning and also provides information related to the network conditions for preparing the service offering matrix (section 3.5.2). This module can be activated after each negotiation or by the physical resource management module, in the case of temporary resource shortages. After processing the dimensioning, it communicates with the physical resource management and the traffic conditioning modules in order to configure the network equipments.

- Physical resource management: The parameters for packet scheduling in each router are dynamically configured for the particular QoS technology, according to instructions, rules and constraints of the network dimensioning. The physical resource management involves the configuration of buffer space (according to the queue management system) and link bandwidth share (according to the service discipline). For example, in a DiffServ network, this module deals with the PHB configuration, according to the service/operation mapping. It is activated whenever there are changes to the resource provisioning, triggered by either a new negotiation or a temporary resource shortage. Among the protocols that can be used for router configuration, a strong candidate is the COPS (Common Open Policy Service) protocol [66].
- Traffic conditioning: It refers to the activities of classification, metering, policing, shaping and marking (for DiffServ), in order to keep the incoming and outgoing traffic according the profiles of the contracted services. Regardless the QoS technology used by the operation plane, traffic conditioning is a necessary task, because the other modules assume that there is some control of the traffic entering in the network. An improper operation of the traffic conditioning can seriously impair the work of the network dimensioning, possibly causing violations of QoS targets.
- Path management: This module is in charge of the aspects related to traffic engineering Path management is an optional module, since not every domain may be willing to change the IP classical hop-by-hop forwarding nature oriented by a shortest-path routing protocol. It manages the routing process according to the guidelines of the network dimensioning and involves pinning routes for helping the QoS maintenance and providing a better utilization of network resources. Traffic engineering can be implemented through a label swapping technology, such as MPLS, or a pure IP solution [203]. The latter relies on existing routing protocols and is undertaken by carefully assigning costs to each network interface [204].

As far as a real implementation is concerned, the same reasoning used for the service plane is valid, i.e., the full set of modules will not be always required. Furthermore, different domains may have different needs, depending on the particular QoS technology they employ. For instance, in the case of a domain employing an approach such as over-provisioning, some modules might be unnecessary. Section 3.7.3 illustrates this situation, by means of an example scenario consisting of some domains with different QoS technologies.

3.3.2 The Resource Manager

The Resource Manager (RM) is the entity in charge of the centralized functions of the operation plane. This definition intentionally excludes those functions typically executed by network elements, including the processing of most protocols. The RM is an integral part of the Service Manager, although it may be implemented as a separate entity.

The RM maintains the operation repository, where the following information is stored:

- Network topology, with network elements (mainly routers), and links connecting them. This information is used by all functional modules of the operation plane. In the case of routers, it includes maximum processing speed, buffer space, scheduling disciplines, particular QoS features (for instance, used PHBs for DiffServ) and vendor-specific features. For links, it includes the capacity (in bps), traffic direction (generally, full-duplex) and underlying transport technology (SDH, ATM, etc.).
- Current configured values of parameters related to queue management, traffic scheduling, service classes and resource reservation mechanisms. This information is strictly dependent of each particular QoS technology.
- Traffic conditioning information, including for instance filters for traffic classification and token bucket parameters used for traffic metering.
- Traffic engineered paths configured by the path management module. For each path, it is necessary to keep information about labels and mapping strategy, for MPLS and link weights, in the case of a pure IP solution [204].
- Historical data for future analysis aimed at tuning the capacity planning module.

3.3.3 Access Networks

The Chameleon architecture deals with various different aspects of end-to-end QoS, although this thesis is focused on service definition and negotiation. The contracted services are expected to maintain their required QoS levels all the way along the path from source to destination, also involving access networks where the end-users are connected. Access domains willing to offer advanced end-user services are responsible for extending the guarantees up to the user premises (corporate user) or equipment (home user). As discussed in section 3.1.3, this is not always easy or even possible, due to the existing characteristics and constraints of each access technology.

For example, the delay is normally very high for dial-up lines, due to the on-the-fly compression done by modems. A modem can add about 100 ms in each side of the communication [96], practically eliminating the chances of two dial-up users to utilize commercial multimedia applications (interactive voice and video application must observe a strict higher bound for delay). An ADSL access network can add about 20 ms, whereas the delay of an Ethernet network is normally below 1 ms. Other technologies, such as ISDN and cable modem also have their own constraints that can influence the performance of certain services. Furthermore, wireless networks (cellular systems, wireless LANs, etc.) present a high packet error rate when compared to its wired counterparts.

The intrinsic characteristics of each access technology must be analyzed before a user service is contracted, in order to provide users the real expectations about the scope of the end-to-end offered guarantees.

3.3.4 Implementation Technologies for the Operation Plane

In Chameleon, the choice of the QoS technology is left as a decision to each domain. In practice, some scenarios involving certain combinations of technologies on access and transit domains are more likely to be deployed in the current Internet.

IntServ/RSVP is a technology indicated for access networks, since the individual flow signaling and state maintenance do not cause problems in relatively small networks. Furthermore, under-provisioning in access domains and in their interconnection with transit domains are considered the most common reason of packet loss in the Internet [160]. Therefore, per-flow guarantees are necessary in some access domains for implementing stricter and more demanding advanced services. In order to assure that the QoS guarantees of IntServ are extended to the access networks, the IETF's ISSLL (Integrated Service over Specific Link Layers) [109] working group has established standards for mapping IntServ to datalink layer technologies including IEEE 802 LANs.

For transit domains, the options likely to be used in any implementation are:

- IntServ: In very limited situations, for small networks or for a small fraction of the traffic, IntServ/RSVP could be used in a truly end-to-end configuration.
- DiffServ: DiffServ was developed for use in the backbone, due to its potential scalability. Currently, in spite of the expectations and hype over it, there is little actual deployment of this technology. The deployment of an overlay network such as Chameleon can give the incentives towards a massive utilization of DiffServ.
- MPLS/TE: Traffic engineering (TE) enabled by MPLS has been used by some providers, in order to gain more control over their network. Telecom operators are strong candidates to use MPLS for implementing their next generation converged networks, i.e., networks that can deploy data and voice on the same infrastructure. TE can be achieved with or without the use of QoS Routing.
- DiffServ over MPLS/TE: The use of DiffServ for providing QoS together with MPLS for traffic engineering is able to yield more benefits than the isolated used of both technologies (although the complexity is higher as well) [76].
- Over-provisioning: Large transit domains (especially Tier 1 providers) are used to keeping their networks with spare resources, in order to get rid of congestions. Consequently, they get considerably low packet loss rates (nearly zero) [160].

When various domains use different technologies, some mapping mechanisms are needed in the borders among them, for maintaining the end-to-end nature of the WDSs deployed by Chameleon. These mapping policies are defined and controlled by the service operation and implemented by the operation plane. In particular, policies and mechanisms for making it possible the identification of the traffic that belongs to the WDSs are needed, as well as the efficient and correct mapping to different technologies that can be found along the end-to-end path (section 4.5). Figure 3.8 illustrates this situation.

The use of IntServ/RSVP in conjunction with technologies such as DiffServ and MPLS requires some changes in the RSVP message processing. The definition of the IntServ Null type of service [26] releases applications from informing their QoS needs. This is useful in the case of an integrated operation of IntServ over DiffServ, for freeing some routers of RSVP message processing. Within DiffServ domains, RSVP messages are processed only on border routers, which map the IntServ class to the proper PHB. Only when they reach the destination IntServ domain, the RSVP messages are processed again. An entire DiffServ domain can be seen by IntServ as just one node in such a scheme, according to RFC 2998 [27]. This architecture maintains the end-to-end feature of IntServ, at the same time it benefits from the scalability of DiffServ.

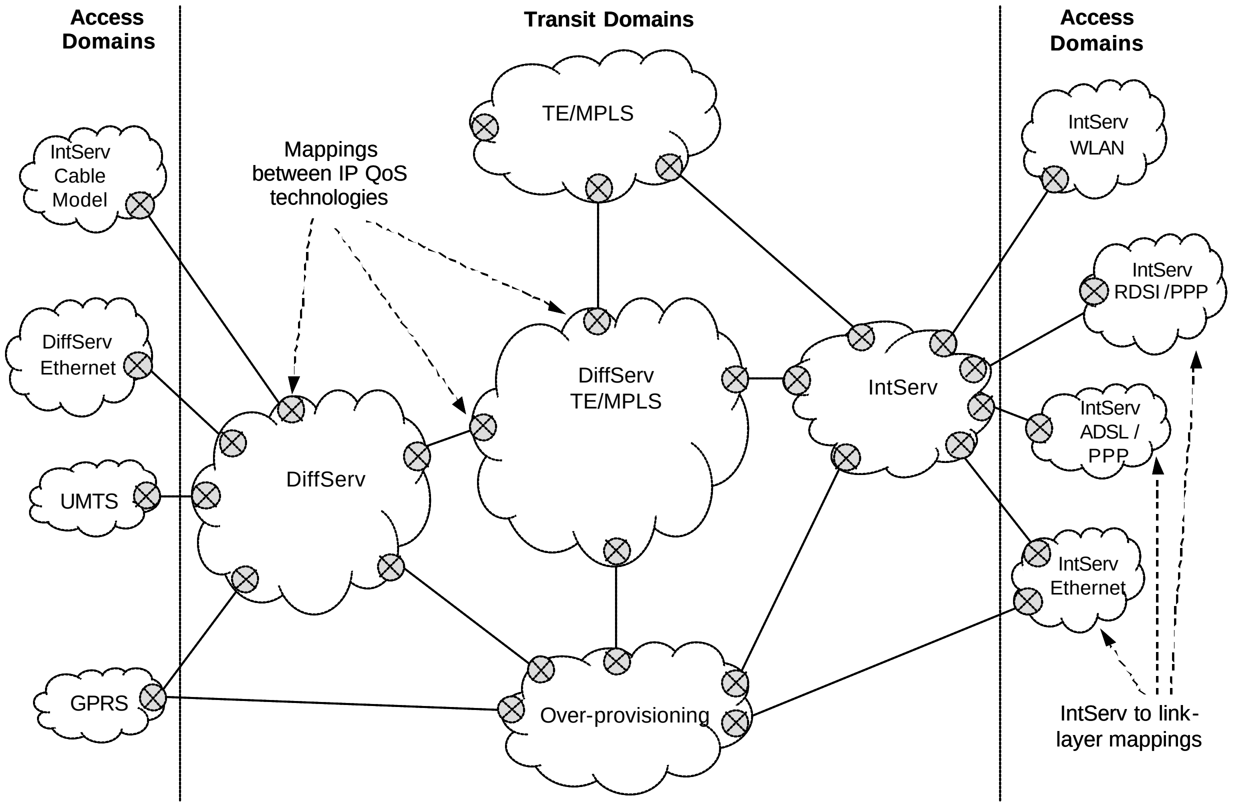


Figure 3.8 – Implementation scenarios with QoS technologies

There are also situations where RSVP messages cannot traverse the end-to-end path (due to firewall policy enforcement, for instance) or when the signaling model determines that RSVP is used only locally. Even in such cases, some applications can get the benefits of resource reservation, by means of the use of a RSVP proxy [87]. This proxy is a RSVP extension that permits intermediate routers to send a RESV message on behalf of the receiver identified in a PATH message. The IntServ Null service type together with the RSVP proxy make it possible a variety of scenarios of end-to-end QoS involving IntServ/RSVP and other QoS technologies such as DiffServ.

3.3.5 End-user QoS Middleware

Middleware has been given different definitions by different people in the last years. In a broader sense, it is anything between the application and the transport layer. Under this definition, the entire Chameleon architecture can be considered as an instance of middleware, specific for QoS implementation. In spite of this broad and loose definition, the discussion in this section is focused on a particular type of QoS middleware that may be required for extending services to the end users.

In its current stage, Chameleon is not concerned in specifying and developing middleware solutions, which are closely linked to the QoS technologies used by each domain. In each possible scenario of Chameleon deployment, different types of middleware may be required in distinct implementation phases. A middleware can intermediate the setup procedures required before the real data traffic is allowed to start, for instance, the establishment of application sessions (SIP or H.323), admission control and resource reservations. Some functions are essential to some QoS technologies (typical end-to-end functions), whereas other ones are only needed for making the transition between legacy best effort applications to a Chameleon-enabled network.

Two proposals for end-user QoS middleware that can be adapted to Chameleon are:

- The EAT (End-user Application Toolkit) [206] from AQUILA architecture aims at providing scalability and efficiency to the transparent support of QoS by multimedia applications. The main motivation behind the EAT is the requirement of enabling the QoS support for legacy applications. Thus, applications not natively developed for AQUILA can also have its benefits. Another goal is to hide the application complexity from the end user, as well as specific aspects of QoS, such as resource reservation, packet marking and traffic specifications. An essential EAT component is the proxy module [207], which is in charge of the establishment of, for instance, SIP and H.323 sessions and RSVP reservations. Since the EAT provides services to the end-user it should be located in access networks, either in every AQUILA-enabled host or in a firewall.
- The QoS Initiator [33] recently proposed by the IETF's NSIS (Next Step in Signaling) [110] working group is responsible for mapping application requirements into QoS service classes, initiating network signaling and activating local QoS provisioning mechanisms. It can be located in the end systems or wherever in the network. This work is just beginning at the time this thesis is being written, though.

3.4 The Monitoring Plane

The monitoring plane is in charge of assuring users and providers that the contracted services are operating strictly within the quality bounds defined in the WDS. It is orthogonal to the service and operation planes, and plays an essential role in Chameleon, by giving the required credibility to the quality of the deployed services. Service monitoring is also considered a factor of success in initiatives such as TEQUILA [16], AQUILA [103], QBone/Internet2 [194], EURESCOM P1008 [56] and TF-NGN [139].

Measurement is the main activity of the monitoring plane, which refers to the process of collecting and storing information for observing a certain characteristic of the measured object. The characteristics of interest are called metrics. Monitoring involves the continuous and repetitive measurement of certain metrics aimed at verifying the existence of predefined conditions (e.g., if the metric exceeds a given threshold).

In order to fulfill the responsibility of keeping services working properly regardless of the network technology of the operation plane, the monitoring plane pursues two objectives: service monitoring and operation monitoring. Service monitoring is related to the service plane and consists of verifying whether the provided services are operating according to their definitions. Service providers may want to monitor their services in order to certify their quality. Service monitoring may be mandatory or optional, depending on the definition of each service deployed by a CDG (section 4.3.1). Not every domain might be able to perform monitoring or some domains may want to do it only for some specific services. Therefore, in case it is mandatory, monitoring influences the choice of whether to deploy or not a service and it can also represent a factor of differentiation, in case it is optional. Users can also be allowed to do the monitoring, in order to have a real time evaluation of the service quality (section 3.4.8). This type of service monitoring is called performance monitoring.

Another type of service monitoring is the traffic profile monitoring, used for checking whether an upstream domain is sending the traffic to the downstream domain in accordance to the traffic profile specified in the WDS. Both domains can monitor the same traffic stream that is crossing their border routers. Traffic profile monitoring can be used just for the evaluation of WDSs or for commercial reasons, such as when the WDS specifies that traffic received in excess should be accounted and charged.

On the other hand, operation monitoring refers to an internal activity of each domain in the operation plane; therefore, it is not mandatory. The goal of operation monitoring is improving the efficiency in resource utilization by providing a timely feedback to some functional modules of the operation plane, such as capacity planning, network dimensioning and path management. Measurements collected for the service monitoring can be used also by the operation management, when they provide the required information.

The extension of the monitoring plane activities may present significant variations, depending on the goals and targets of the domain. As far as scope is concerned, monitoring can be seen as an activity to be performed only within a domain (section 3.4.5), among domains (section 3.4.6) and even between the end systems involved in a communication (section 3.4.7).

3.4.1 Functions of the Monitoring Plane

The functions of the monitoring plane include those directly related to the measurement process or derived from it, set to achieve service and operation monitoring. Figure 3.9 depicts the functional modules of the monitoring plane and their relationship with the measurement repository, where all the information collected and processed is stored:

- Measurement: This module collects metrics of interest and stores them in the measurement repository. In addition to fulfilling the monitoring objectives, measurements are required also for two important activities of the service plane: traffic prediction and measurement-based admission control. The measurement process is influenced by some factor, such as metrics of interest, measurement purpose, measurement frequency, underlying technology used by the operation plane, and the availability of measurement tools.
- Analysis: It is responsible for the post-processing of the collected information. The range of activities of the analysis module varies according to the goals of each domain and the purpose of the measurements. It includes, but is not restricted to: computing statistics (mean, standard deviation, etc.); filtering of non-significant information in order to reduce the amount of data stored in the repository; and tendency analysis.
- Alarm generation: Alarms are generated when certain thresholds are exceeded, such as the violation of a delay guarantee for a given service. This function is coordinated by domain policies and can be done in real time, as a direct result of the measurement process, or based on stored data. Alarms are sent to both service and operation planes,

hence redirecting the activities of service negotiation and network dimensioning, in order to keep the network working in proper conditions. This module operates in a timescale that varies from fractions of seconds to hours.

- Performance reports: This module is responsible for generating analysis reports about service and operation performance, in a medium-term timescale (or days or weeks). It is useful in providing insightful information for strategic decisions related to the service negotiation and resource provisioning.
- Metric exchange: Metric exchange is the activity whereby domains exchange some metrics of interest collected and analyzed previously, in order to perform interdomain monitoring (section 3.4.6). It can be implemented by means of a simple request-reply protocol through the M-M interface.
- Traffic statistics: This module is dedicated to formatting information (computed by the analysis module) and replying the requests from the traffic prediction module of the service plane.
- Load evaluation: It is the responsible for maintaining up-to-date information about current network load and replying to requests of the admission control module of the service plane. Load evaluation is only required when a domain is implementing a measurement-based admission control (MBAC) scheme.

Note that these functional modules represent a comprehensive set of activities related to the monitoring plane. Similarly to the service and operation planes, domains are not expected to implement all functions of the monitoring plane. This depends on the requirements imposed by the CDG, as well as on the individual domains’ capabilities. An example scenario with the required modules of the monitoring plane is presented in section 3.7.4.

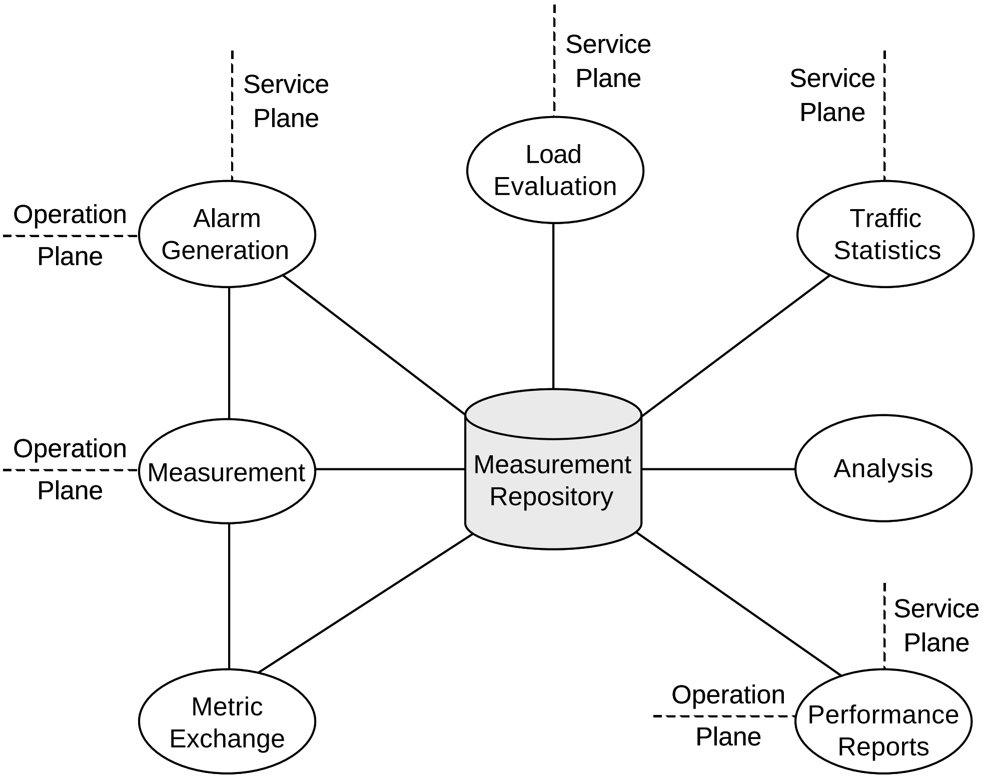


Figure 3.9 – Functional modules of the Monitoring Plane

3.4.2 The Monitoring Coordinator

The functions of the monitoring plane are executed by the Monitor Coordinator (MC), which is a part of the Service Manager (section 3.1.1). Its implementation is comprised of a variety of SNMP managers for collecting metrics, as well as investigative tools (probes) and protocols that operate at the M-M interface. The MC maintains the measurement repository that stores the following information:

- Metrics of interest for performance and traffic profile monitoring collected in active or passive measurements (section 3.4.3).
- Historical information about alarms that were generated and sent to the service and operation planes.
- Current load in every link, for MBAC.
- Per path network state (from an ingress to an egress router) for every service, which may include throughput, delay, jitter and packet loss, depending on the service.
- Performance reports sent to the service and operation planes.

3.4.3 Metrics and Measurement Tools

Domains must agree with respect to the metrics to be monitored, before end-to-end advanced services that require active monitoring are successfully deployed. Often, metrics will be specified by the SLS part of the WDS, although this is not a mandatory feature. In any case, metrics must be well defined (no ambiguities), in addition to be quantified and measured by existing, accessible and affordable measurement tools and devices. The implementation of Chameleon recommends the use, whenever possible, of metrics standardized by the IETF's IPPM (IP Performance Metrics) [108], including the proposed terminology [161].

Metrics can be roughly categorized into two groups, depending on the type of monitoring they are intended to achieve. Performance metrics are refinements from the parameters used for specifying QoS guarantees in the SLS (section 4.3.1). Some examples of performance metrics are [56][139]: available bandwidth (for a given service); used bandwidth; one-way delay [7]; one-way delay variation [60]; and one-way packet loss [8]. Performance metrics may be used both for service and operation monitoring. Traffic profile metrics are used for characterizing the traffic flow between two neighbouring domains. Examples of these are data rate, maximum burst size, number of packets per time unit, and packet size [56].

The procedure for collecting metrics may vary according to its type. For example, collecting a given metric may or not impact the network operation. Under this criterion (called transparency), measurement can be classified as: active or passive: Passive measurement is performed without impacting existing traffic or network elements operation, because it does not injects additional traffic to get the samples. Passive measurement can collect all traffic crossing a given point in the network or only a fraction of it. It is always possible to collect data automatically stored in MIBs (Management Information Base) by network elements. Some metrics are adequately collected by passive measurements, such as available bandwidth, current data rate and maximum burst size.

Active measurement is performed by injecting traffic probes (test packets) in a point of the network and verifying its properties in another point. The disadvantage of active measurement is that the packets artificially injected in the networks may influence the measurement accuracy, since they represent additional traffic. It may also contribute for increasing the network load, if the measurement frequency is excessively high. The artificial traffic injected by active measurements must faithfully represent the real traffic generated by the applications that use the monitored service. Some metrics are only obtained by active measurements, such as delay (one-way and round-trip) and delay variation.

There is presently a variety of different tools for measuring performance and traffic profile metrics, both through active and passive measurement [35][56][139]. Some protocols also have been proposed (for intradomain or interdomain monitoring), such as the one-way active measurement protocol (OWAMP) [180]. Despite the rich diversity of tools, the work is only beginning and most tools are not yet complete and/or they do not present accurate results. Some of them do not have a clear definition and only a few are being standardized. This is due to the fact that traditionally the Internet offers the single best effort service that does not require any sort of measurement for working properly. It is also difficult to find accurate tools for measuring high-speed links. Similarly, presently domains do not rely on measurements for accounting and billing, because the capacity of the underlying interconnection link determines the amount of data a user (or domain) can fill in. With the growing interest for advanced services and the consequent need for achieving performance targets and charging based on network utilization, measurement tools are expected to experience a significant development in the next few years.

3.4.4 Measurement Points

Measurements are strategically performed in some points of the network, where the collected information is more meaningful. These measurement points can be located in the end systems, as well as in some other places [56]. Figure 3.10 shows two users accessing an advanced service in a User Service Provider and the potential measurement points spread over the end-to-end path. In this scenario, the end systems are connected to the Internet through dial-up and broadband ADSL access links.

Three types of measurement points can be identified in Figure 3.10:

- Endpoints (E): These measurement points are located in end systems, such as, servers, personal computers, laptops and palmtops. Extending the monitoring up to the end systems may no always the possible, since it depends on the availability of end-to-end monitoring (section 3.4.7).
- Intermediate points (I): These points will most likely be used for both interdomain and intradomain monitoring. They are located at domain borders, most commonly in

routers (or devices coupled to them). Intermediate points can be further classified as ingress and egress points.

- Access points (A): They represent a particular case of intermediate points, yet important enough for deceiving a separate classification. Access points are located in concentration points, where the traffic of a large number of (home) users is multiplexed into their access domain.

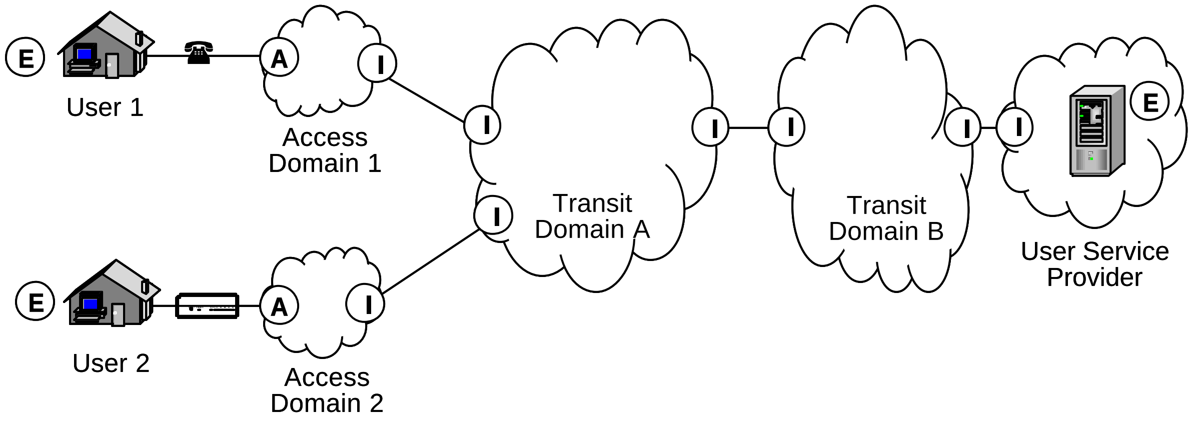


Figure 3.10 – Measurement points (E – endpoint; I – intermediate; A – access)

The identification of these points is relevant, principally in the context of interdomain monitoring, which naturally requires standard procedures and interfaces. In the forthcoming sections, this classification will be used in the context of intradomain, interdomain and end-to-end monitoring.

3.4.5 Intradomain Monitoring

The scope of intradomain monitoring is limited to the inside of domains. Operation monitoring is a form of interdomain monitoring, since the operation plane is only visible within domains. A great deal of the activities of service monitoring is also characterized as intradomain, including the activities of collecting, storing and analyzing metrics of interest, and also generating alarms and reports. The objects to be monitored by the intradomain monitoring function are network elements, interconnection links and internal paths [16][56] (in the case of a traffic engineered network). In this case, the measurement points are pairs of I points. The communication with the operation and service planes is through the internal interfaces.

3.4.6 Interdomain Monitoring

Interdomain monitoring happens between two measurement points located in different domains, aiming at verifying the adherence of the network operation to the requirements of the contracted services. Particular monitoring protocols are needed for exchanging metrics[14], through the M-M interface.

Domains take part in the interdomain monitoring in an active or passive mode, i.e., by initiating investigations or by responding to requests initiated by other domains. A domain can initiate an investigation for preventive or corrective reasons. Preventive (or proactive) monitoring is typically initiated by user service providers for guaranteeing the quality of their offered services. It is carried out regularly, with a variable frequency (section 3.4.9). The goal of corrective (or reactive) monitoring is identifying the cause of a problem when it is detected and fixing it. A manual process can initiate the corrective monitoring, according to some criteria defined for each domain and the experience of the service administrator. It can also be initiated in an automated way, for getting additional information about a problem that was identified in the preventive monitoring or triggered by some alarm generated by the end-to-end monitoring.

The participation of the measurement points in the interdomain monitoring can be of two forms: direct or hop-by-hop. Each one requires specific monitoring protocols. The direct case involves only the initial and final intermediate points (for instance, the nearest I points to the end systems, in Figure 3.10). This type of monitoring is recommended for preventive investigations, whereby knowing the service behaviour between measurement points is desired, due to its higher precision and lower interference with the network. However, it has some limitations. Firstly, sometimes it may not be possible to do it. It is expected that a good deal of domains will deploy advanced services (given the number of existing administrative domains in the current Internet), making it more complex for every domain to carry out interdomain direct monitoring to each other. Secondly, when a problem is detected, direct monitoring does not give any information about the network point where it is happening.

Hop-by-hop monitoring consists of examining collected and stored metrics, among two or more consecutive measurement points. Two drawbacks of this approach are that it can retrieve stale information and that domains must necessarily trust the information given by each other.

[14] It is outside the scope of this thesis to specify monitoring protocols. Possibly, one single protocol will not be enough, due to different monitoring styles and objectives. One example is the One-way Active Measurement Protocol [180] for measuring the one-way delay metric, currently being defined by the IETF's IPPM working group [108].

An intermediate approach is a real time hop-by-hop monitoring, involving information exchange among various consecutive measurement points.

3.4.7 End-to-end Monitoring

End-to-end monitoring happens between end systems, in order to investigate whether the QoS guarantees of the user services[15] are being respected. In Figure 3.10, end-to-end monitoring could be performed between E measurement points.

Although it might seem interesting always to carry out e2e monitoring in order to keep control over the user service performance, in practice it can cause some problems. The systematic use of end-to-end monitoring as a mechanism for evaluating the user service quality is not realistic, since it depends on the installation and maintenance of dedicated monitoring software in all end systems. Such investigation can be done using statistical sampling, by choosing some representative users (similar to the current strategies of measuring audience indexes for TV channels).

Furthermore, end-to-end monitoring may lead to false interpretations in some cases. The main factors that can affect the measurement result are access networks, which are based on different technologies. In some cases (such as Ethernet or cable modem), the sustained QoS levels depend on the network load, since all users share the same network capacity. Although there are mechanisms for guaranteeing QoS for these technologies, frequently providers cannot interfere on networks that they do not own (e.g., a corporate network or a access network that belongs to a telecom operator).

3.4.8 User Service Management

One typical situation wherein the end-to-end monitoring may be necessary is when the user service offers the facility of User Service Management (USM) [210]. The availability of the USM adds value to the user service, so that it should be defined in the user SLA. Although highly related to user services, the USM has a huge end-to-end implication and therefore cannot be considered out of the scope of the Chameleon architecture.

The concept of USM can be illustrated more easily by the availability of a GUI for the end user, which provides information on the service quality and permits the user to do limited

[15] Both intradomain and interdomain monitoring refer to transport services.

interventions. The functionalities provided by a particular implementation of the USM depend on the service provider and on the service itself. The information on the service quality is obtained form the end-to-end monitoring and it can be shown to the user in the GUI, through, for instance, traffic light signals representing good (green), medium (yellow) and poor (red) quality. Additionally, the user may be requested to inform whether or not he/she is happy with the service quality, in order to provide an efficient tuning of the service. Two examples of the user interaction with the USM are:

- The user has a service with some QoS guarantees, which are not absolute guarantees, though. That is, at moments of congestion the service is not able to sustain the same quality, although it performs better than the best effort service (this service is frequently referred to as Assured [151] or Olympic [102] service). In such moments of low service quality (according to the user perception), he/she can improve the quality through a "quality gauge" on the GUI. The provider can implement this functionality by increasing the throughput share for this service in the access network or by changing the mapping from this user service to a higher quality transport service. Obviously, the user may incur a higher cost, and he/she should be aware of it.
- The user has a service that assures low, but variable, packet loss and delay. For some applications, such as video on demand and radio broadcasting, the QoS parameters could be configured to a higher value. For stricter applications, such as interactive audio and video, the user could lower both loss and delay by some control at the GUI. Such behaviour could be implemented by changing the mapping between user and transport service.

The availability of end-to-end monitoring is essential in those cases, in order for the provider to obtain a proof that the user is receiving the quality he/she is paying for.

When it comes to implementation of the USM, it can be based on some existing QoS middleware, such as the EAT (section 3.3.5) from the AQUILA project.

3.4.9 Measurement Frequency

The measurement frequency for a given monitoring situation involves a number of factors and depends on the accuracy, additional network load and volume of retrieved information [56]. More frequent measurements are able to more faithfully represent the service levels delivered by the network. However, this approach can adversely affect the performance of links and routers and can generate a huge amount of information to be stored in the repository. In the case of active measurements, the frequent injection of additional packets can harm the network performance. Passive measurements do not generate traffic by themselves, but they require the information to the transmitted to the Monitor Coordinator, though. They also can waste excessive CPU cycles of routers for measuring and storing metric samples. This situations becomes more drastic at high-speed links; For instance, an OC-192 core link (10 Gbps) demands much more frequent measurements than a 256 Kbps access link.

The decision for choosing the measurement frequency for service and operation monitoring has some differences. For the former, the frequency is defined by the service itself and specified in its formal definition (WDS). For the latter, the frequency is defined according the internal policies, generally a compromise among accuracy, performance, volume of information and measurement goals.

3.5 Resource Control in Chameleon

Resource Control is a functional module of the Service Plane that is responsible for collecting enough information about resource availability and demand from the Operation and Monitoring Planes in order to help the service negotiation process. It also deals with resource provisioning aspects in a technology-independent way, interacting with the Operation Plane for performing the physical provisioning in the QoS technology chosen by the domain.

The module of Resource Control in Chameleon is divided up into three different other modules: Resource Estimation, Service Offering and Resource Provisioning. In turn, these modules interact with the Traffic Prediction and Admission Control modules, which are also commented in this section.

3.5.1 Resource Estimation

This functional module determines the amount of resources a domain will request to other domains in future service negotiations, which will be needed for guaranteeing the end-to-end resource provisioning. Each time a domain wants to buy services (section 5.1.3), it needs to estimate the resources that will be needed for each domain and each service. The outcome of the Resource Estimation is a resource request matrix, shown in Figure 3.11, which will be used by service buyers when participating in service negotiations. The content of each cell selects only one domain and one service and its unit is given in bps. It may increase or decrease compared to the previous one, depending on the information and algorithm used for resource estimation. Additionally, for those destination domains that declared its internal PoPs to the CDG (section 3.1.2), each cell must contain a list of PoPs and the resource requirements for each one (obviously, the sum of them must be the same of the domain aggregate resources). This is useful for the domain's internal resource provisioning.

Destination	**Service**		
	S1	**S2**	**S3**
D2			
D3			
D4			

Figure 3.11 - Resource estimation matrix

Domains may consider several factors for performing the Resource Estimation function:

- Traffic Prediction (TP): The analysis of past traffic is the basic information used for resource estimation. When traffic volume presents a steady behaviour, TP generally is able to catch up with the variations, producing estimates that are higher than the real traffic. On the other hand, when traffic volumes suffer abrupt variations, a good estimate may require other factors (listed below) seen as modifiers to the value generated by the traffic predictor.

- Admission Control (AC): For those services that are subject to admission control, some calls/sessions might be blocked. In these cases, no traffic is generated. Therefore, in order to allow the blocked users to be admitted in the future, the traffic that would be generated by them must be included.

- Contract Fluctuation (CF): New users that have recently subscribed to a service will generate an additional amount of traffic. Conversely, users that have unsubscribed will not generate traffic any longer. The resource estimation module can use this information for increasing or decreasing the amount of resources to be requested, or alternatively let the traffic predictor to consider these variations.

- Statistical Multiplexing Gain (SMG): This information is important when considering traffic that may be generated by new users and blocked calls, given that the particular service allows some violations of the QoS guarantees for obtaining a substantial statistical multiplexing gain.

- Temporal Variation (TV): Traffic volumes from different applications present variability on different timescales, e.g., hours of the day, days of the week and months. In these cases, historical data may be used to predict trends [65], generating a table of indexes that serve as multipliers to the basic predicted resources.
- Over-Provisioning Factor (OPF): This factor indicates how much a given domain is willing to spend in order to improve the end-to-end guarantees for a service. In a certain way, the over-provisioning factor represents a measure of the domain's disbelief on the other domains and on the accuracy of the previous factors.

Predicting the future is always an error-prone activity. However, domains may use the above factors to get a reasonable idea of the upcoming traffic. The following equation represents just one way how the factor may interact for producing a good estimate:

$$ER = OPF \times (TV \times (TP + SM \times (AC + CF))) \qquad 3.1$$

where ER represents the final estimated resources. The factors TP, AC and CF represent a traffic unit, whereas OPF, TV and SM are multipliers. Factors AC and CF must be normalized to be used in this equation. When $OPF = 1$, $TV = 1$ and $SM = 0$, $ER = TP$, which is the basic factor for resource estimation.

3.5.2 Service Offering

Service offering is an activity that must be done for those domains that want to sell services. For each offered service, a domain must have the following information:

- The internal paths that will carry the traffic of that service. A path is a unidirectional route between a pair of routers, which are two border routers (BR) in transit domains, and a border router and an internal router (IR, that gives access to the end-users or connects to web-hosting companies) in access domains. A domain may even have more than one path between the same pair of routers, provided it is able to have control over the traffic forwarding through different paths. How domains implement their internal paths and the topology is a matter of local interest only, because it is hidden from the other domains in the operation plane. Obviously, for having control over network paths some kind of label swapping technology, such as MPLS, is necessary.

- The performance guarantees that will be provided in each internal path, given by the QoS metrics of the service definition (section 4.3.1), i.e., delay, jitter, packet loss and throughput (throughput is the metric that is called resource in this context). This information is obtained from the operation and monitoring planes through the S-O and S-M interfaces[16].

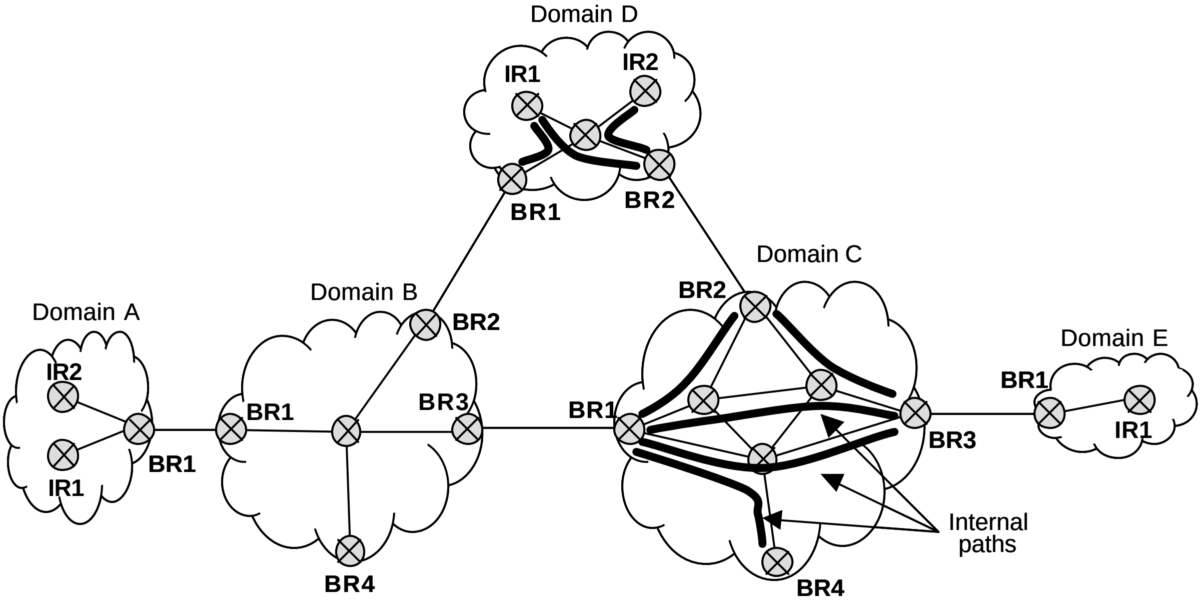


Figure 3.12 – Domain's internal paths

The choice of which services to offer, in which paths and their associated performance guarantees, depends on both technical and commercial aspects and it is up to each domain to decide. Figure 3.12 depicts a small CDG with five domains (three access and two transit domains). The outcome of the service offering module is the service offering matrix. An example for domain C of Figure 3.12 is shown in Figure 3.13.

Domains may or may not be obliged to expose the service offer matrix to other domains (or to any other outside entity), depending on the negotiation model used by the CDG (section 5.4).

Some domains may have tens or hundreds of such border or internal routers. This may produce very large matrixes. In this case, a better approach is to aggregate internal paths by PoP: one BR or IR represents one PoP.

[16] The particular way each domain implements its operation and monitoring planes in order to obtain this information is not discussed here.

Internal Path		Service				
		S1		S2		
From	To	Loss	Throughput	Delay	Jitter	Throughput
BR1	BR2					
BR1	BR3					
BR1	BR3					
BR1	BR4					
BR2	BR3					

Figure 3.13 – Service offering matrix

3.5.3 Resource Provisioning

Provisioning refers to the determination and allocation of resources needed to implement services at various points of the network [25]. In the operation plane, resource provisioning involves both physical and logical activities. Physical provisioning refers to the procurement of network elements and the order of links (new ones or upgrades) and is related to capacity planning. Logical provisioning refers to the configuration and tuning of the network elements (e.g. routers) in order to provide the negotiated performance guarantees to the services. It is related to network dimensioning, physical resource management and traffic conditioning.

In the service plane, resource provisioning is related to the decisions involving the amount of resources to be allocated to each service in the operation plane, independently of the QoS implementation strategies. It is responsible for controlling the provisioning in the operation plane through the S-O interface. After the confirmation of the service negotiation (either sale or purchase), each domain must perform its internal provisioning in order for end-to-end services to be used.

Resource provisioning is highly simplified, since service scope was limited to the pipe style (1:1). Therefore, each domain can map the end-to-end service provisioning into a pipe comprised of one ingress router and one egress router. This is applicable to both service purchase and sale.

3.5.4 Traffic Prediction

Traffic prediction is the most important factor influencing the resource estimation. The ability to predict the future network capacity utilization is essential for service negotiation in

Chameleon. The aim is to forecast future traffic variations as precisely as possible, based on the measured traffic history. Traffic prediction gives some interesting characteristics to the resource provisioning, including: a) stability: resource provisioning can be reconfigured within some predefined time intervals, as a result of periodic negotiations (section 5.4.4); b) scalability: traffic aggregates may be used for predictions.

The traffic samples are requested by the traffic sampling module of the service plane to the measurement statistics module of the monitoring plane and stored on the traffic repository (Figure 3.6). Traffic measurements are performed by the measurement module of the monitoring plane on egress routers. The nature of the traffic samples depend on the prediction model adopted (see below) by each domain. The traffic prediction module retrieves the traffic samples from the traffic repository.

Four aspects are of particular interest for traffic prediction:

a) Prediction scope: Prediction is always end-to-end, i.e., from a source to a destination domain, as depicted by Figure 3.11. In other words, transit domains do not perform predictions on traffic that simply crosses them. A related topic (though it is a responsibility of the monitoring plane) is how a domain doing predictions identifies the traffic being forwarded to the destination domain. Each domain must know the IP address range of each other domain in a CDG. Hence, each network participating in advanced service deployment must use IP addresses within the range of a CDG member domain.

b) Traffic identification: This falls under the responsibilities of the monitoring plane, but it is related to the traffic prediction in the service plane.

c) Prediction frequency: There is a tradeoff between the length of the prediction interval and prediction error. The larger the prediction interval (i.e. the lower the frequency) is, the less accurate the prediction becomes. Therefore, small prediction intervals are preferred for achieving small prediction errors.

d) Prediction model: A variety of models has been proposed for traffic prediction in computer networks in recent years. One of the simplest is the Local Gaussian Predictor [44][65] used in previous evaluations of the Chameleon Architecture [129][130][133]. More sophisticated models include Auto-Regressive Moving Average (ARMA) model and the Markov Modulated Poisson Process (MMPP) model [177].

A very important point is the predictor's ability in producing good estimations. Two problems may arise, namely, overestimation and underestimation. Underestimation should be completely avoided, since it may cause contract violation, as some of the QoS parameters are not fulfilled. On the other hand, ove restimation should be minimized, in order to avoid the provisioning of too many resources that will not be used at all. These problems may be caused by a too large prediction frequency and/or an inappropriate prediction model.

3.5.5 Admission Control

In networks that support advanced services, such as Chameleon-enabled networks, admission control is necessary to determine whether a new traffic flow (call/session) can be admitted to the network while current applications maintain the QoS performance guarantees they require to work properly [135]. For the TEQUILA project, admission control is considered a key component for QoS deployment, because it "determines the extent to which network resources are utilized and whether the contracted QoS characteristics are actually delivered" [147]. Admission control is an activity associated with user services. Depending on the application characteristics and service QoS guarantees strictness, user service may be subject or not to admission control. Real-time applications, such as interactive multimedia, teleimmersion and action network games typically are subject to admission control, while adaptive applications, such as web browsing and file transfer, are not.

There is no admission control for transport services, which are the focus of the Chameleon Architecture. Once the resource provisioning was performed as a result of the service negotiation, the resources are made available. However, admission control affects the resource control of transport services in some ways:

- Blocked calls/sessions may influence the resource estimation, as discussed in section 3.5.1.
- By comparing admission control statistics with resource estimation values, it is possible to discover whether the negotiated resources are being underestimated or overestimated.
- There are two possible methods of performing admission control for end-to-end services, with respect to the scope: local or end-to-end. Local admission control is the preferred one, because it just checks the availability of resources within the source domain. On the other hand, end-to-end admission control is performing by recursively

exchanging messages from source to destination domains. In each domain, resource availability is verified. Since it is triggered by each service invocation, it raises concern related to its scalability[17]. Hence, the goal is to use local admission control, provided that the end-to-end resource provisioning is correct.

Admission control is logically performed by the Service Broker, although a real implementation may be distributed. In AQUILA, responsibilities are divided among a Resource Manager Agent (RMA) and some Access Control Agents (ACA) [153]. It is out of the scope of this thesis to propose mechanisms for dealing with admission control in Chameleon.

3.6 Chameleon versus Over-provisioning

Over-provisioning may be seen as a means for obtaining QoS in the Internet (section 2.3.6). In this case, a question that may arise is whether or not the deployment of a more complex overlay network such as the Chameleon architecture will be necessary. Three questions must be answered:

1) Is over-provisioning able to provide QoS performance guarantees?

2) If this is true, what is cheaper: a QoS technology or over-provisioning?

3) In the case of over-provisioning being cheaper, is it enough for providing end-to-end QoS? Since a domain has to rely on some other domains to deploy an end-to-end service, can it trust these domains? In other words, can a domain trust that other domains will keep their networks over-provisioned all the time, in order for advanced applications to meet their performance requirements?

It is the view of this work that some of the answers for the above problems may be:

1) Over-provisioning may not be enough for providing guarantees for the more demanding applications [24]. Over-provisioning is just another QoS deployment approach and as such the same reasoning is applied. It may be suitable for deploying some services but not necessarily for *every* service. For instance, voice traffic requires a transport service with low delay and minimal jitter an packet loss, although its throughput need is low (a conversation can be compressed to 8 kbps). Even though a network is over-provisioned,

[17] Similarly, to when using IntServ, although it is not so drastic, since it is not hop-by-hop.

if packets are delayed in queues when very short congestions occur, it may not be able to fulfill the QoS needs for voice applications at these times.

2) It may be too expensive to get performance guarantees by relying solely on an over-provisioned network. Common intuition says that as long as the aggregate rate in any link is maintained low (e.g., less than 50%, i.e., an over-provision factor greater than 2) the worst-case delay within the network will be low. However, even this intuition may fail under certain network conditions [41]. For guaranteed services dependent on worst-case delay bounds, over-provisioning may either be too expensive or not work at all. In any case, this is a choice that each domain has to make based on the type of services it wants to offer, deployment costs, previous experiences (if available), scientific work and the opinion of experts and the feeling of the technical staff.

3) Domains should not trust in other competing domains unless a SLA is established between them. If there is no SLA, a domain cannot be sure that the traffic will be forwarded all the way from source to destination with the required QoS performance guarantees. In such a case, what determines the guarantees is not the over-provisioning, but the SLA. Therefore, a "negotiation platform" will be needed, for dynamically negotiating services, and there will be room for the Chameleon's deployment.

3.7 Implementation Scenarios

The Chameleon Architecture has been designed on the basis of planes and functional modules, because it was originally conceived as an extensive framework, representing a wide variety of options and choices. However, when it comes to put it to work in a real scenario, this broad spectrum of possibilities needs to be narrowed to a simpler system, in order for the implementation to be feasible. For this reason, this section presents an illustrative example of some possible implementation scenarios for the Chameleon Architecture, concerning a small CDG starting its activities. In these scenarios, some functions for the service, operation and monitoring planes (presented in sections 3.2, 3.3 and 3.4 respectively), are either deliberately excluded or combined into simpler ones.

Moreover, as described in section 3.1.1, related functions from different planes cannot always be clearly distinguished and separated into different software entities in a real implementation. Therefore, in this example some functions from different planes are implemented together. For this simple example scenario, the Service Manager is supposed to be implemented as a single logical entity. They are not necessarily bundled into the same program, but are part of a single system, specifically developed for that purpose. The exceptions are some existing measurement and management tools used for undertaking the activities of the monitoring plane.

Along this section, forward references to Chapter 4 (service definition) and Chapter 5 (service negotiation) are made, whenever concepts and definitions that have not yet been defined are used.

3.7.1 CDG Structure

The CDG of this sample scenario is comprised of seven domains, including three Transit Providers (TP), and four Access Providers (AP), as shown in Figure 3.14. Transit Providers employ different QoS technologies: TP1, TP2 and TP3 use Traffic Engineering based on MPLS, over-provisioning and DiffServ, respectively. Access Domains employ either IntServ or DiffServ. However, each one uses a different sub-IP technology in its access network: cable modem, ADSL, Ethernet and dial-up connections. Access Provider 3 is a private corporate network; the other ones are ISPs.

Cooperative deployment of common advanced services is the main goal of the CDG. Each domain must sign a multi-lateral SLA, whereby it agrees on maintaining the negotiated service levels for the traffic belonging to the other members. The function of CDG Controller (section 3.1.2) is played by one of the members (i.e., there is no third-party entity in charge of the CDG).

As a starting-point, the CDG has defined two advanced services to be deployed, following the idea of the Two-bit Differentiated Services proposal [151]. In Chameleon, however, they are defined by means of Well-Defined Services (WDS, section 4.1):

- Premium: a quantitative service for interactive multimedia applications, characterized by throughput and one-way delay as performance parameters. Premium service is a simplification of the Conversational Multimedia Service, defined in section 4.4.2.

- Assured: a qualitative service that provides better-than best effort quality [25], primarily intended to fast web browsing.

All domains must deploy at least the Assured service in order to be accepted as CDG members, whereas deploying the Premium service is optional. In this example, access provider

4 chose not to deploy it at first, because it would not be able to extend the service scope to the access networks (section 3.1.3 presents a discussion about end-to-end service scope; in general, dial-up networks have limitations in extending performance assurances to end users).

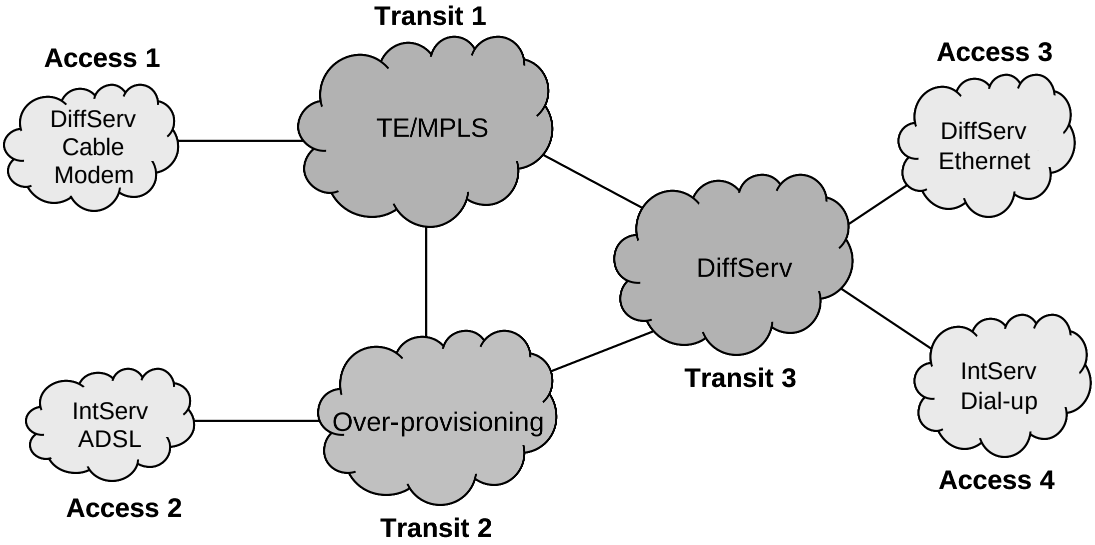


Figure 3.14 – CDG topology and technologies for the example scenario

The next sections present some details of the implementations of the service, operation and monitoring planes.

3.7.2 Service Plane

The service plane in Chameleon is responsible for creating the virtual service overlay network, which coordinates the activities related to service definition and negotiation, and resource control (section 3.2.1). The functional modules described in section 3.2.1 were proposed to achieve a broad set of possibilities selectable for a given scenario. In this example, the requirements of the CDG can be implemented by a subset of those functions, depicted in Figure 3.15.

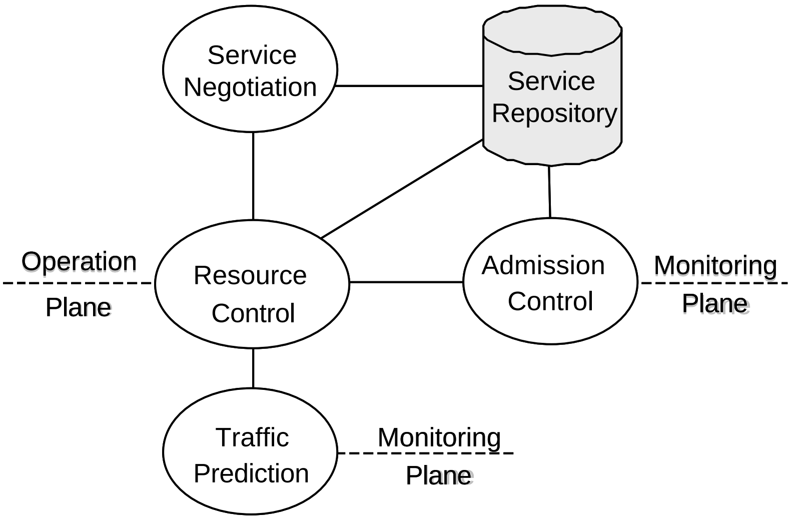


Figure 3.15 – Modules of the service plane for the example scenario

While domains are free to implement the service plane functionality at their own discretion, this section presents some basic recommendations of the CDG that can also be used by all domains. Since the CDG has a collaborative approach, domains are likely to adopt the same structure for the service plane.

- Service negotiation: As described in Chapter 5, the main design choice for the service negotiation module is an appropriate negotiation model. In this example, since the CDG is small, the members agreed that the Cascade negotiation model (section 5.5) would be used, implemented by a modified version of the SIBSS protocol [194]. The three transit domains act as service sellers, whereas the four access domains act as both service buyers and sellers.

- Resource control: the three functions related to resource control (estimation, offering and provisioning) are necessary for implementing this example.

  - Resource estimation: this function is implemented only by access domains (service buyers), which build a resource estimation matrix based on traffic prediction and admission control.

  - Service offering: the underlying network topology and the share of link capacity allotted to each service are configured in a static manner, in order for this module to be able to prepare the service offering matrix. The amount of throughput to be offered to each service (Premium and Assured) depends on particular goals and policies of each domain. The CDG recommends that no more than 10% and 30% of the capacity be offered at each link for the Premium and Assured services, respectively. Domains obtain the information of the one-way delay they are able to provide to the Premium service by means of the OWAMP protocol [180], used at the monitoring plane. The service offering matrix is built as presented in Figure 3.13, i.e., comprising paths connecting border (or internal) routers. In order to determine whether routers and links are up (resource availability), the service offering module implements a small function of the corrective reaction module: it probes routers via the SNMP management protocol.

- Resource provisioning: this module is activated as a result of service sale or purchase and its implementation depends highly on the particular technology of the operation plane. Therefore, in this case the best implementation strategy is bundling the resource provisioning module together with some modules of the operation plane.

- Traffic prediction: A simple Local Gaussian Predictor [65] is recommended by the CDG (which is also used in the simulation study of performed in Chapter 6, described in section 6.1.2). This predictor was chosen because it is simple but effective, as demonstrated in [44][65], and also in section 6.2.
- Admission control: The Premium service requires admission control. The CDG recommends the use of a MBAC CAC method [121], based on the measurements performed by the load evaluation module of the monitoring plane. Although the guarantees provided by such a method are not strict, it permits higher resource utilization (through multiplexing) than peak rate CAC, and do not incur in the complex formulation of a statistical CAC [162].
- Service repository: the service repository, together with the operation and monitoring ones, are implemented "all-in-one" by a single public domain DBMS, such as MySQL and PostgreSQL, with SQL access. Another option could be a LDAP protocol accessing a directory system. However, it shows little flexibility when extending its bases for reaching new applications. Some benefits of the DBMS approach are scalability, integrity and concurrent access [80].

The functions performed by other modules (not included here), are described next:

- Service definition: services are statically configured by the service administrator according to particular policies defined for each domain and the set of WDSs deployed by the CDG.

- Service/Operation mapping: this function was simplified by some static configurations stored on the service repository for mapping services into QoS technologies, as described in section 3.7.3.
- Corrective reaction: it is taken by the service administrator, by observing reports generated by the monitoring plane.
- Routing control: the cascade negotiation model (adopted for this example) is not able to interfere in the normal routing computation. Therefore, this module is not used.
- Domain discovery: addresses of SBs of other CDG members are manually configured.
- Historical indexes: this function is not supported.
- Traffic sampling and traffic repository: not included, since the traffic prediction module receives samples directly from the monitoring plane.

3.7.3 Operation Plane

The functionalities provided by the operation plane must be carefully customized for each domain that employs a different QoS technology (unlike the functions for the service plane). The general approach, recommended by the CDG, is to implement the modules of this plane together with the resource provisioning module of the service plane.

The subset of modules needed in this example scenario and their applicability in each domain are described in this section. As Figure 3.16 points out, the modules of the operation plane are:

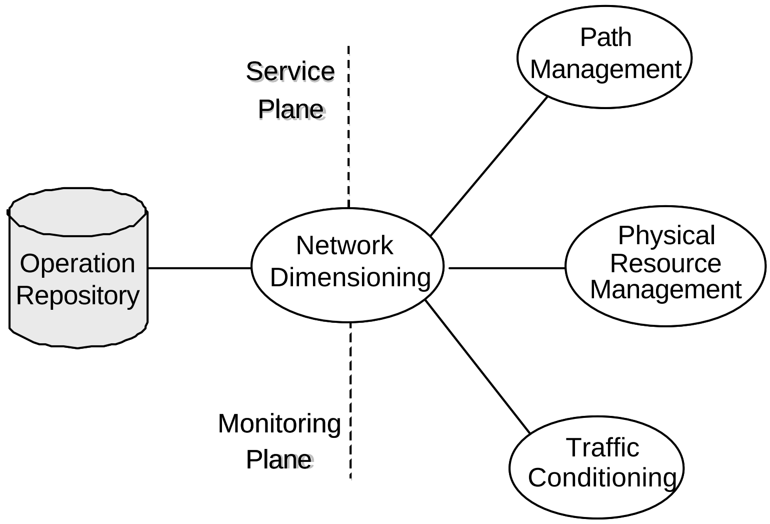


Figure 3.16 - Modules of the operation plane for the example scenario

- Network dimensioning: translates a resource provisioning matrix yielded by the service negotiation into the resources of the underlying network for accommodating the generated traffic levels. It is needed by MPLS and DiffServ domains. Strategies for network dimensioning proposed for the TEQUILA project [205] can be used for this example scenario of a Chameleon implementation. However, unlike the general proposal of section 3.3.1, for the sake of simplicity it does not receive feedback information from the monitoring plane in the form of alarms and reports. The administrator is informed of relevant events by a traditional network management software (logically located in the monitoring plane) and takes the appropriate corrective actions.

  This module is not needed by IntServ domains, since requests for resource reservation are launched on-demand. OP (over-provisioning) domains also do not implement it, because in this case there is no need for bandwidth management. As other functions of the operation plane are necessary for both types of domains, the network dimensioning module can be preserved for compatibility purposes, but operating in a pass-through transparent way.

- Path management: this module is only needed by the MPLS domain, since the other domains are not implementing alternative techniques of Traffic Engineering. After having accommodated the resource provisioning matrix, the network dimensioning module sets the LSPs by means of the LDP protocol [11]. The LSPs are supposed to be pre-configured by an off-line Traffic Engineering system [20] in order to accommodate both Premium and Assured services. This module is mostly intended for small changes in order to adapt dynamically to traffic fluctuations, by means of redirecting traffic to other backup LSPs. The network dimensioning module can be configured with some policies concerning throughput and delay thresholds for accepting a given traffic aggregate to be forwarded through a LSP. The overall goal of these policies is keeping the network utilization evenly distributed.

- Physical resource management: this module is responsible for configuring the scheduling and queuing mechanisms at routers. This function is primarily intended for DiffServ domains, in order to dynamically reprovision the bandwidth share allocated to each PHB. IntServ may also need this module, for configuring thresholds for accepting resource reservation requests. The implementation is performed by the COPS protocol, with this module at the Resource Manager as the Policy Decision

Point (PDP) and routers representing Policy Enforcement Points (PEP). A variety of different DiffServ parameters can be configured through COPS, as defined by the DiffServ Policy Information Base (PIB) [81].

- Traffic conditioning: all domains implement this module at border routers. Traffic conditioning is necessary even for domains implementing over-provisioning, since they should not submit traffic to their downstream domains above the negotiated levels. In the case of DiffServ domains, the need is obvious, since it is supported by the idea of traffic conditioning applied at border routers and PHB enforcement at core routers. For MPLS domains, traffic conditioning can also be useful for the FEC-to-NHLFE mapping. The configuration of traffic conditioners at border routers is also performed by the COPS protocol.

Network planning is not implemented by any domain in this example scenario. The module responsible for capacity planning is performed off-line by the network administrator, for simplicity reasons, since it sees the long-term network requirements. The functionalities associated to network engineering are also not supported since they demand a specially designed network infrastructure to become operative.

In addition to the general recommendations for using the functional modules of the operation plane, each domain must be concerned with some given particular configurations:

- Transit Provider 1: the implementation of TE/MPLS does not support Constraint-based Routing paths (CR-LSP [123]). The Premium and Assured services are mapped onto LSPs specially engineered so that the congestion level is controlled. The Premium service is mapped to paths with lower edge-to-edge delay, for keeping up with the service definition requirements and the delay limits offered in the negotiation.

- Transit Provider 2: no particular remarks, since over-provisioning does not require particular traffic control mechanisms. Even though, monitoring is essential for guaranteeing the QoS levels agreed in the service negotiation.

- Transit Provider 3: The DiffServ implementation of the Premium service is mapped to the EF PHB [59] and the Assured service to the AF PHB group [102], with respectively 10% and 30% of the link capacity (controlled by careful provisioning). The traffic conditioning action for out-of-profile Premium packets is “discard”. The Assured service uses one AF class (AF1) and two levels of discard priority. In-profile and out-of-profile packets are market to AF10 PHB and AF11 PHB, respectively.

Discard priorities for the AF PHB are dealt with by using the RIO [49] scheme (a RED [83] queue management mechanism that is able to discriminate in and out packets). PHBs are implemented by a Weighted Round Robin (WRR [84]) scheduling mechanism.

- Access Provider 1: DiffServ implementation is similar to that of Transit Provider 3. The extension of the Premium and Assured services through a QoS enforcement mechanism at the access network is performed by mapping DiffServ into cable modem QoS support [95], according to the DOCSIS [141] specification.
- Access Provider 2: IntServ over ADSL uses the ISSLL [109] mapping of IntServ into PPP [116]. RSVP/IntServ scope extends until the last internal router (egress router), by means of a RSVP *Proxy* (section 3.3.4). Both Premium and Assured services are mapped into the IntServ controlled load service [217], because the requirements for implementing the guaranteed QoS service are too strict [182].
- Access Provider 3: DiffServ implementation is similar to the Transit Provider 3. The extension of the Premium and Assured services to the access switched Ethernet network needs some sort of mapping from DiffServ to the IEEE 802.1p standard [82].
- Access Provider 4: implements IntServ over dial-up lines. The Premium service is not supported by this domain. The Assured service is mapped to IntServ according to the ISSLL recommendation (similar to Access Provider 2). IntServ is mapped to DiffServ according to RFC 2998 [27].

In general, the mappings between each pair of QoS technologies follow guidelines presented in sections 3.3.4 (implementation options for the operation plane) and 4.5 (WDS mapping, based on a WDS identifier).

3.7.4 Monitoring Plane

Monitoring is a very important activity in order for a domain to provide others with the necessary confidence that the negotiated service levels are being met and that possible service outages will be detected as soon as possible. This is true even for Transit Provider 2, which relies solely on over-provisioning. In other words, the fact that it does not implement any particular technology for guaranteeing the agreed performance assurances does not exempt it from constantly verifying the effective service level that traffic is experiencing as it traverses the underlying network.

The subset of functions proposed for the monitoring plane in section 3.4.1 and implemented in this example scenario are shown in Figure 3.17. There are some differences between access and transit providers, with respect to the metrics of interest. This section does not attempt to describe such particular aspects of each domain, though.

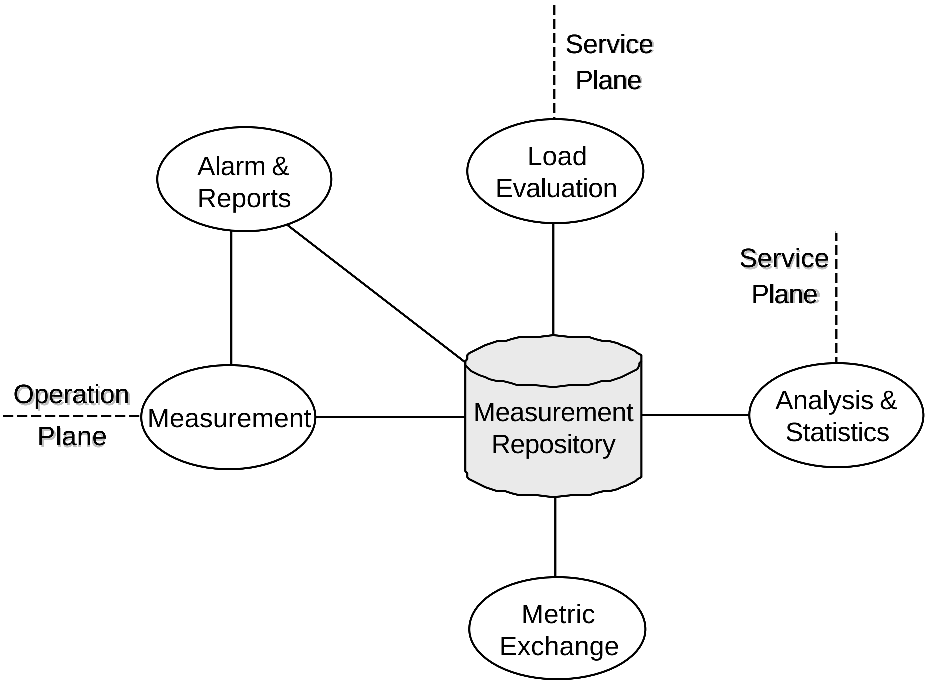


Figure 3.17 - Modules of the monitoring plane for the example scenario

- Measurement: both active and passive measurements are performed, by beans of different measurement tools. Active measurements are needed for monitoring metrics such as the one-way delay; whereto a good candidate is the OWAMP protocol [180], which is able to capture the one-way delay metric defined for IPPM [7]. There is a number of tools available for measuring one-way delay, such as the Active Measurement Tool (AMT) [124]. In the same way, the available capacity of a network path that can be used for MBAC is measured by active measurements, such as in the *pathload* tool [120]. Passive measurement is needed for getting the effective throughput provided for Premium and Assured services. Along with other metrics for intradomain monitoring (packet loss, queue length), this metric is measured by means of SNMP queries to router MIBs. Information collected by measurement tools is stored in the monitoring repository.

- Alarm and Reports: this module combines both alarm generation and performance reports modules proposed in section 3.4.1. Provided that the main metrics of interest are collected, this module continuously compares them with the negotiated values, taken from the service plane. In case of the Premium service, the comparison is achieved via simple thresholds, whereas for the qualitative Assured service, the comparison is against the best-effort service, via queue length and packet drop statistics at routers. Alarms are generated by traditional network management tools. Performance reports are also generated by a network management software, or simpler software for generating statistics, such as MRTG and RRDtool [156]. Both alarms and reports do not generate feedback for the service and operation planes (because it would require a much more complex implementation, and possibly generate system instability). Reactions to service outages are taken manually (off-line) by the network administrator.
- Metric exchange: interdomain monitoring is performed by the fictitious Simple Interdomain Monitoring Protocol (SIMP, section 4.4.2), for exchanging throughput and delay information for the Premium service.
- Load evaluation: this module provides information for the admission control module of the service plane, based on MBAC. Domains of the CDG can choose between one of the two following approaches: a) collecting traffic samples (peak, average, and standard deviation) from all routers involved in the implementation of the Premium service, via the SNMP protocol; b) using the abovementioned *pathload* tool.
- Analysis and statistics: it represents the combination of two modules, aimed at producing mean and standard deviation of traffic samples for the traffic prediction module of the service plane.

3.8 Summary

The high level design of the Chameleon architecture was presented in this chapter. Chameleon gives a step forward to the current proposal for Internet QoS by clearly separating the functions particular to abstract services from their implementation with QoS technologies. The service and operation planes are logically isolated from each other, and the communication between them happens through well-defined interfaces, following the usual approach of isolating layers in network architectures. This separation makes it possible for a domain to choose any approach for obtaining QoS, as long as it is able to provide services with the required performance guarantees. The deployment of transport services is completed with the monitoring plane, which collects information from the operation plane, presents some analysis and feedback relevant results to the operation and service planes for on-the-fly service tuning.

The proposal of a brand new architecture was not strictly necessary for exposing the main ideas of this thesis. They could come as extensions for the existing TEQUILA, AQUILA or CADENUS architectures, which share some common features. However, all of them (also Chameleon) have been developed simultaneously with the same goal and from the same background, but with slightly different approaches. Therefore, the contributions of this work are best fitted in the Chameleon architecture, although they could be implemented also in the other architectures. The design of Chameleon is also a contribution.

The next two chapters are exclusively devoted to the most important functions of the service plane: service definition and negotiation.

Chapter 4

Advanced Service Definition

Service definition within the context of the Chameleon Architecture refers to the activity of precisely explaining the semantics of a service[18]. This definition allows all domains to understand service requirements in the same way, so that a unique end-to-end behaviour may be achieved, no matter how many different domains are cooperating to deploy it, or the sort of different underlying QoS technologies these domains are using. This notion of services with known and well-documented features and behaviours is paramount to Chameleon. The approach for defining advanced services described in this chapter is one of the contributions of Chameleon. This architecture goes to a great extent and level of details so far not found in other similar approaches in order to achieve service definition.

In the sections that follow, the concepts involving the definition of QoS-based advanced services are elaborated. Section 4.1 introduces the concept of Well-Defined Service (WDS), which is how services are explicitly defined in Chameleon. The influence of SLAs in Chameleon and how they are related to WDSs are the subject of section 4.2. In section 4.3 the concepts of WDS classes and instances are presented, as well as their format and content. An example of both a WDS class and instance is given in section 4.4. Section 4.5 is concerned with clarifying how packets belonging to a given WDS can be identified by different domains, taking into account that they can use different QoS technologies. As the concept of WDS was influenced by (and also had some influence on) other similar approaches, section 4.6 presents related work. Section 4.7 summarizes this chapter with the main features of service definition in Chameleon.

[18] As opposed to giving a definition of the word “service”.

4.1 Well Defined Services

In the Chameleon Architecture, service definition is based upon the concept of Well-Defined Service (WDS), which is a service that has a clear and unambiguous definition of the performance guarantees that a provider offers or wants to receive when a SLA is being negotiated. It must exhibit the same behavior in every domain where it is implemented, to make it possible to deploy an end-to-end service to end-users. In this context, service negotiation refers to transport services (section 2.2.4), defined and negotiated by domains. It is expected that end-user services, including voice and video, are negotiated beforehand, independently of the transport service negotiation. User services must be mapped to transport services in order to enable end-to-end service negotiation and utilization.

WDS is a new level of abstraction, because it allows Internet services to be something that can be set up according to applications and users' needs. Today's Internet best effort service is not a service by itself, but it is just an outcome of the IP protocol service model. Along with service negotiation, the concept of WDS enforces the view that Chameleon's service plane should be seen as a separate plane.

WDSs are for the deployment of advanced services in the Internet the same that IP is for systems connectivity: the unifying glue. For the IP protocol, the same protocol is spoken end-to-end in the Internet, regardless of the underlying transport network technology (called sub-IP) used by given domains. The same rationale is used for WDS. Given that a Chameleon Domain Group "speaks" (i.e., deploys) a certain WDS, it may be used as a basic support for deploying advanced applications (that need QoS guarantees), independently of particular QoS technologies used by domains for implementing it.

WDSs do not need to be standardized. Rather, a CDG simply has to agree on a WDS in order to deploy a new service. The main idea is that some WDSs will become very popular and then they may become standardized. It is also important to stress that not every CDG member needs to implement every WDS available in its CDG. Domains are given the freedom to choose the WDSs they want to deploy (for any reason).

With regard to scope, WDSs may be categorized into two types: Globally Well-Known WDSs (WKS for short - Well-Known-Service) and Local WDS (LS - Local Service). There is no difference between them other then that WKSs are intended to be used in an Internet-wide scope. Eventually, an LS may become a WKS, if it is accepted by a large number of CDGs.

4.1.1 Motivation for WDSs

The adopted service definition mechanism has a considerable influence on service negotiation, provisioning and monitoring. As a result, these aspects are considered and thoroughly examined in Chameleon. There are a number of different approaches for service definition:

- By means of common or well-defined services (Chameleon’s approach).
- Using standard SLSs that specify parameters, which can be filled with values for creating a service during the negotiation.
- No service definition is used. This approach makes sense when network over-provisioning is used as an alternative for QoS technologies. The influence of over-provisioning in Chameleon is discussed in section 3.6.

Next, some benefits resulting from the use of WDSs, as opposed to letting each domain understand and implement the service semantics as faithfully as it can., are presented:

1. WDSs make service negotiation easier, by limiting parameter and value combinations.
2. They help in preserving the end-to-end service semantics and guaranteeing a correct implementation of these semantics by all domains of a given CDG. It is difficult for a domain to capture the semantics of a service and all its implications on the implementation simply by observing some parameters in the SLS during negotiations, as proposed in [93]. It has been already recognized that “the mapping process between the generic SLS and the concrete QoS mechanisms can be very complex if the user can freely select and combine the parameters” [175]. Furthermore, some parameter combinations make more sense than others.
3. Relying on each domain’s interpretation of the service semantics may lead to a situation where successive mappings do not preserve the original service. Even if standard SLSs format and negotiation protocol are available, an end-to-end service definition is still necessary, and it serves as a basis for the utilization of more elaborate service negotiation models. Clearly, WDSs may be seen as important players when it comes to activities related to service deployment, such as, resource provisioning, network dimensioning and traffic engineering.

4. It is commonly expected that, for the sake of simplicity, backbone networks will deploy a limited number of QoS traffic classes [175]. In such a scenario, even though the service definition scheme allows having several different parameter configurations in the SLS, actually there will be only a few available WDSs.
5. Service monitoring is far easier when service features and requirements are well known by every member of a CDG.

Summarizing, Well-Defined Services are a guarantee for service buyers and sellers that the service is sufficiently well understood by all of them, so that they can rely on the end-to-end service performance guarantees.

Different service features and network scenarios may influence the correct service implementation. The choice and combination of QoS parameters needed to describe a service may lead to different ways of configuring the network. Some possible combinations are: throughput and delay; throughput and jitter; throughput, delay and jitter; throughput, delay and packet loss. In addition, these parameters themselves may have different interpretations in the context of different services. Certainly, it is possible to trade flexibility for accuracy, presenting well-defined definitions of all performance parameters in the SLS format in order to avoid ambiguities. For example, an end-to-end Virtual Wire (VW) service, based on the Virtual Wire PDB [115] could think of jitter as being “phase jitter” (as in the VW PDB definition) contrary to the most common definition of “inter-arrival jitter”. Should a standard definition, such as IPPM IPDV [60], be adopted for jitter, there would be no problem in understanding the parameter. However, the deployment of the VW service might become no longer feasible. In some cases, even the network topology can affect the guarantees offered by a service and a domain that implements such a service should be aware of such influence. For example, although the difference in specifying a determinist service and a predictive service might be in the form a percentile for QoS parameters, such as delay and jitter, both require a different implementation.

The point here is that if a domain can accurately understand a service during an SLS negotiation by simply observing its parameters, then actually there is only one type of service that may be parameterized according to some particular interests. This service should satisfy all possible variations that end-users and other domains could ever desire. Since this is not likely to be true, this approach will only work for very simple parameter combinations.

IntServ services are formally defined and identified. The two services defined by the work group, Guaranteed Quality of Service and Controlled Load, are different indeed. If both were considered variations of a same service, probably just one base service with parameters to differentiate them would have been standardized. It may be difficult (if not impossible, in some cases) to capture the semantics of a service only by the QoS parameters. For example, the Controlled Load service states that: “the transit delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum transmit delay experienced by any successfully delivered packet” (RFC 2211 [217], section 2). Of course, it would always be possible to change this definition for a quantitative one, such as: “99.5 % of the packets will experience a maximum delay of 100 ms”. Another option would be a qualitative definition: “most packets will experience low delay”. Both definitions try to approximate the service to fit in an established SLS format, but they do not necessarily catch the original service semantics in the same fashion.

In some other scenarios, the SLS negotiation process may become quite complex to capture the desired service semantics. For example, a client could ask the network for the lowest possible delay, instead of specifying the maximum accepted delay. In order to obtain such a service, one could define an SLS format and create a more complex negotiation protocol (or another mechanism). However, this implies a trade-off between flexibility in service specification and simplicity in service negotiation. Another simple solution would be prohibiting this kind of service.

The use of WDS is a step towards dealing with all the above problems. They are a way of specifying a single behaviour for an end-to-end service, distinguishing semantically different services that may demand different implementations. However, a WDS itself remains independent of the implementation mechanisms adopted in a particular domain. In this sense, the WDS concept fully satisfies Chameleon’s plane separation concern, stated in section 3.1.

4.2 SLAs in Chameleon

Chameleon is intended to deploying end-to-end services possibly involving many domains, which have to agree on some services beforehand. Consequently, an SLA is naturally multilateral in Chameleon. Furthermore, as defined earlier in section 3.1.2, a CDG is a very broad and loose concept. The number of domains in a CDG may vary from as little as two domains to as high as the scalability of the negotiation model allows. The liability and duties of each member domain also may vary considerably from one CDG to another.

Similarly, the terms of an SLA may differ substantially from one CDG to another, and the Chameleon architecture may be deployed by entities ranging from a truly collaborative context (a group of universities, such as the Internet2) to a real competitive environment. In the latter case, the contractual "teeth" (financial repercussion for not matching the SLA performance conditions [154]) are likely to be a very important point, whereas in the former they may even not be necessary at all. Therefore, the focus in Chameleon is on defining the SLA's technical part, in other terms, the WDS.

Nevertheless, one very important question to be answered is related to the responsibility for the service deployment from a business relationship point of view. Who is responsible for the deployment of a WDS? To whom the buyer domain (section 5.1.3) will complain, in the case of a service outage? This not only depends on the CDG targets but also on the chosen negotiation style and model (section 5.4). In an end-to-end negotiation style, the responsibility is distributed over every member, and buyer domains must be aware of this situation. In such a scenario, the deployment of well-structured monitoring capabilities is very important for resolving performance problems and outages. Even though domains are under a strong competition, they must collaborate in order for advanced services to be properly deployed. On the other hand, in a border negotiation style it is more commonly assumed that the downstream domain (which is the service seller) be the responsible for the service.

Regardless of the negotiation model being used, the CDG Controller is responsible for keeping track of established SLAs and WDSs of a given CDG. Its functions include maintaining a WDS repository, with active WDS classes and instances (section 4.3).

4.3 WDS Classes and Instances

Instead of attempting to define all possible Transport Services as WDSs, a better approach would be to group them into classes, each of which encompassing semantically similar services. From these classes, new service instances can be created through careful parameter configuration. A service class should be defined in a formal document and each domain that adheres to offer services based on this class must be able to implement it correctly. The advantage of having WDS Classes is to facilitate service negotiation by having a limited number of service categories (or groups), wherein several different WDSs may be classified. WDS classes help in maintaining small the number of active services negotiated by a CDG. Domains may instantiate services filling their associated parameters with particular values in order to customize services to their requirements. Within each class, services can be dynamically instantiated through parameter negotiation. Optionally, providers can create permanent instances of WDSs (with their own identifiers) using the class identifiers and specifying pre-defined values for some parameters. Note, however, that this is merely a way of organizing services and simplifying the negotiation through SLSs.

The information for specifying a WDS Class is divided up into different sections. Each section has a requirement level, which may be “mandatory”, “conditional” or “optional”.

- Service Identification (mandatory)

  Services are identified by means of a WDS identifier (WDSID) and an auxiliary human-readable service name. Domains specify the desired WDS during negotiations using the associated WDSID, which has an end-to-end meaning, but does not interfere in the way traffic belonging to a certain WDS is identified within these domains (see section 4.3.2).

- Service Description (mandatory)

  In this section, the service provided by the WDS class is described, in order to provide insight on the service semantics and its intended use.

- Service Level Specification (mandatory)

  Traditionally, SLSs are only for unidirectional traffic. In Chameleon, the SLS may additionally define a service with bidirectional traffic (section 4.3.1).

- Formal Proof (conditional)

  Mathematical proof, depending on the class and the level of guarantees provided by the service (e.g., whether it is a guaranteed or a predictive service)

- Attributes and Constraints (optional)

  Additional attributes considered valuable for clarifying some aspects of service behaviour and constraints of its use. This information may help different types of domains participating in a CDG, such end-user, access and backbone service providers. For instance, constraints may be used to limit the scope types (section 4.3.1) or negotiation models (section 5.4) for the present WDS class.

- Examples of Use (optional)

  Contains examples of how this service may be used for supporting end-user services and also hints on mapping end-user services into this WDS class.

- Guidelines for Implementation (optional)

  Despite the fact that WDSs are abstract services and thus they are independent of the actual implementation, a WDS Class may provide some information of the way it could be implemented using different QoS technologies.

The SLS should have a unique format for all classes, in order to avoid unnecessary complexity. When a client requests a service of a class that it knows beforehand the provider is able to deploy, it has a guarantee that the end-to-end service semantics will be maintained.

4.3.1 SLS Format and Content

Traditionally, services specifications are unidirectional, for instance, IntServ services or services build by using DiffServ PHBs. By the same token, the TEQUILA's SLS proposal [93] states that, "*an SLS is associated with unidirectional traffic flows. Note however that this does not exclude the provisioning of bidirectional technical agreements, by combining one or more SLSs.*". In contrast, a Well-Defined Service in Chameleon necessarily defines a whole service. In other words, a WDS may be comprised of one or more SLSs, each of which dealing with one direction or level of QoS performance guarantees. For example, if a given end-user service is intrinsically bidirectional, as in IP Telephony, the WDS for supporting it should be bidirectional as well. In other cases, different functionalities that require different QoS guarantees may be bundled together in a user service by the service provider. For example, a video on demand service may need different levels of guarantees for the video stream and control functions (stopping, pausing, resuming, etc.). One way of forwarding control traffic is through the best effort service. However, some providers may consider that this traffic needs some guarantees, but not the same strict guarantees offered to video traffic. Moreover, video traffic is unidirectional, whereas control traffic is bidirectional.

The SLS format of a WDS is comprised of some different sections, having as a basis the above-mentioned TEQUILA's SLS format[19]. This choice was made because TEQUILA's approach is the most comprehensive SLS format developed so far. It was derived from a

[19] A more in-depth description of the SLS sections and parameters can be found in [93].

seminal work developed as a draft for the DiffServ working group [25], which introduced the term SLS, and defined almost all information needed by Chameleon's WDS specification. Both TEQUILA and Chameleon architectures have been developed in parallel, since the year 2000[20].

- Scope (mandatory)

  Different scope styles, distinguished by the number of ingress and egress domains, may be supported in Chameleon, including Pipe (1:1), Hose (1:N), Funnel (N:1) and Generic (1:any and any:1) [93]. Service negotiation and resource provisioning are easier and even more precise for the Pipe style because of its one-to-one nature. This thesis assumes that all service scopes are of the pipe style, unless otherwise stated. The difficulties in using non-Pipe styles in a multi-domain environment go beyond those found in resource provisioning in QoS-enabled hose-style VPNs, as reported in [63].

  The service scope is represented by a pair of network addresses, for identifying source and destination domains. Depending on the negotiation model, the scope may also be represented as a list of such pairs (e.g. in the hierarchical model, section 5.7).

- Service Schedule (optional)

  Indicates when the service will be available, that is, the start and end times. It is an optional parameter and its default value is "forever", meaning that resources will remain reserved unless the service is explicitly deactivated (by means of requesting "zero" resources for it in a negotiation).

- Reliability (mandatory)

  Indicates the duration a service is allowed not to be operational per unit of time (month, year), in a unit such as the mean downtime per year (MDT). In case of an outage, it also indicates the maximum allowed time to repair (MTTR).

- Monitoring (optional)

  The WDS class may stipulate whether the service will be subject to interdomain monitoring (that occurs through the M-M interface) and the rules and metrics associated with it. The monitored metrics are the parameters belonging to the

[20] Actually the author of this thesis participated (via discussion list) with valuable suggestions in the development of TEQUILA's SLS format, as it is expressed in the acknowledgment section of the formal document [93].

"performance guarantees" section of the SLO (see below). Additional information of the monitoring section may be:

- Monitoring frequency: The frequency of the interdomain measurements, according to discussion in section 3.4.9.
- Monitoring protocol: Defines the protocol for monitoring this particular WDS. It does not exclude the possibility of defining a standard interdomain monitoring protocol, though.
- Data expiration time: When a domain receives a monitoring request, the measurement does not necessarily have to be performed right away. The domain is allowed to inform the value of a metric that is stored in the measurement repository, as long as it is still valid, according to the data expiration time.

- Service Level Objective – SLO (mandatory)

  A SLO [216] is a part of the SLS intended to identify the traffic and the QoS performance guarantees that must be provided for it. A SLS may be comprised of one or more SLOs, each of which dealing with a specific treatment that must be applied to (part of) the traffic. Two or more SLOs in a SLS can be used for:

  - Dealing with bidirectional services. In such a case, one SLO is needed for each direction.
  - Applying different QoS guarantees for parts of the traffic that make a WDS, which may be, for instance, user and control information.

  An SLO is made of some components:

  - Traffic Identification

    Traffic identification is auxiliary information to the WDSID intended to provide a more fine-grained view of the traffic composition inside a WDS. It is defined by the tuple *<IP source address(es); IP destination address(es); protocol(s); source port(s); destination port(s)>*, which represents a filter. Each element of the tuple can be assigned to single value, a group of values, or a null value. The latter means that the filter component will not be used.

- Traffic Profile

  Traffic profile is a set of parameters describing characteristics of the traffic which performance guarantees must be applied to. Traffic profile monitoring (section 3.4) is performed based on this SLO component. Most common parameters are transmission rate (expressed in bps) and maximum burst size (expressed as a percentage of the transmission rate). Other parameters may include maximum allowed packet size and number of packets per unit of time. The exact number of parameters and their definitions depends on each WDS class.

- Excess Treatment

  This SLO component specifies how a downstream domain will process excess traffic, i.e., traffic that does not conform to the traffic profile. Excess traffic may be dropped, shaped (until it conforms to the profile), accounted (for future billing) or remarked (when using a packet marking technology, a.k.a. DiffServ).

- Performance Guarantees

  Refers to the QoS performance guarantees provided to the packets of this SLO within the scope of the WDS class. These are limited to transport services and do not include necessarily access networks (not strictly end-to-end). An access provider may use different technologies (section 3.3.3), each of which having different characteristics with respect to performance guarantees. Therefore, access providers may limit the scope of the WDS to their internal networks, actually limiting the access network to the scope of the end-user service.

  There are four main performance parameters, whose exact meaning must be defined by the WDS class.

  - Delay: This parameter may refer to either one-way delay [7] or round-trip delay [9], at the discretion of the WDS class. Most quantitative services rely on one-way delay, whereas the round-trip delay is often sufficient for qualitative ones. A case in the point is the TCP protocol, whose performance is mostly influenced by the RTT (Round-Trip Time) [77]. Measuring RTT is easier and cheaper than one-way delay.

  - Jitter: Traditionally, jitter means different things for different people. Recently, however, the concept of delay variation of two consecutive packets has been used instead of jitter, as defined by IETF's IPPM working group (IPDV [60]). In this work, the term jitter is maintained, since each particular WDS class may give a different definition for it.

- Packet loss: It is a ratio between the number of lost packets from source to destination domains and the amount of packets submitted by the application in the source domain. This definition expresses the idea of one-way packet loss, although some WDS classes may prefer a round-trip packet loss parameter.
- Throughput: It is defined as the bit rate coming out the last hop of the WDS scope in the destination domain. In some cases, when the destination domain presents to the CDG its internal PoPs, the throughput must be represented as both an aggregate value for the whole domain and a list of individual throughput values for each individual PoP. Throughput may or may not be considered an independent parameter, because depending on the definition of the traffic profile it can be calculated as a function of transmission rate and packet loss [93].

The ways the above parameters will effectively be represented in a WDS class depend on the level of guarantees of the intended service. On the one hand, qualitative services may express delay, jitter and packet loss as loosely as "low", "medium", "high", and throughput as "higher than best effort". In such a scenario, service monitoring will be very difficult or even impossible to implement, as it is based on subjective judgments of service users.

On the other hand, quantitative services will most likely specify for these parameters values that will be measured in a specified time interval. For delay and jitter, unless a very strict guaranteed service is intended, a percentile will also be given. A WDS class may specify a parameter as a scalar value or as made of levels or ranges. For instance, a delay parameter for a voice service may allow 2 levels, each of which having different percentiles. The WDS instance may specify values for it; say 150 ms and 400 ms, with 150 ms defined as the default value.

This scheme gives the negotiation more flexibility. When requesting a service, domains may specify that they want a certain parameter with "any level above the 3rd level", or "at most the 2nd level". Obviously, this kind of parameterization depends on the specific QoS levels provided to the user service.

- o Direction: Specifies the traffic direction which the SLS will be applied to, with respect to domain sources(s) and destination(s) of the service scope. Creating a bidirectional service involves two SLOs, each of which covering one direction.

4.3.2 Service Parameter Types

A WDS has different types of parameters, which may be distinguished by their functions, strictness and time when they have to be filled up with values. Some parameters, including those of sections "Monitoring" and "Performance Guarantees" receive real values at WDS instantiation time. On the other hand, parameters of sections "Scope", "Flow Identification", "Service Schedule" and "Reliability" are instantiated at negotiation time. In some cases, WDS instances are created at negotiation time, resulting in a compression of both phases.

- Definition parameters: The values of the definition parameters are specified at the time of defining WDS classes or instances. The CDG Manager stores these parameter values together with the WDS definition.
  - Class parameters: Most parameters can be defined at WDS definition time, except for the scope and throughput of the performance guarantees section. Delay, jitter and packet loss may or be defined together with the WDS class.
  - Instance parameters: Delay, jitter and packet loss of the performance guarantees section may be chosen to be defined at the actual instance creation (permanent or not). The WDSID, for a permanent instance is also seen as a definition parameter.
- Negotiation parameters: The values of this type of parameters are only defined at negotiation time, both for helping the service purchase or sale. They can be part of the WDS definition or only auxiliary parameters.
  - Purchase parameters: Scope and throughput are parameter values that are given for those domains interested in buying services at negotiation time.
  - Sale parameters: Depending on the service negotiation model (section 5.4), information about service sale must be shown, in the form of a service offering matrix (section 3.5.2). This information is not part of the WDS definition.
  - Protocol parameters: Parameters used by particular negotiation signaling protocols, for carrying partial results of delay and throughput, for instance. They also do not belong to the WDS definition.

4.4 An Example of a WDS Class and Instance

The work with service definition in Chameleon is not intended to produce any WDS Class specification, but just to provide a template for service definition based on the WDS concept. The definition of specific service is to be a task for CDG members, when dealing with local services, or even for standardization forums, such as IETF or ITU-T, when considering global services. Global WDSIDs should also be assigned by IANA [107].

4.4.1 Quantitative and Qualitative WDSs

Initially a very small number of transport WDS classes needs to be defined, since providers should be able to map several customer services into a few transport services. The WDS classes can be broadly classified as offering quantitative and qualitative services (section 2.2.4). Examples of quantitative WDS classes are:

- Class 1 - Virtual-Wire: this is a guaranteed service aimed at emulating a dedicated circuit. It can be used anywhere there is a desired to replace dedicated circuits [115].
- Class 2 - Conversational Multimedia: it is a service for dealing with conversational multimedia applications, such as IP telephony and videoconferencing. This is described in details in sections 4.4.2 and 4.4.3.
- Class 3 - Streaming Multimedia: this is a service for dealing with streaming multimedia applications, such as Video on Demand (VoD) and transmission of radio. Streaming multimedia is not so stringent in terms of delay, because it is not interactive.
- Class 4 - Shared Virtual Reality: this is a service intended to support teleimmersive applications, which sometimes are called the ultimate QoS-critical application [136].

Examples of qualitative WDS classes are:

- Class 5 - Assured: this is a service that gives a not quantifiable assurance that the traffic will experience a performance better than the best effort service in times when the network is congested [25][151].
- Class 6 – Olympic: the Olympic service is basically a multi-assured service, with three levels of assurances: gold, silver and bronze [93][102].

- Class 7 - Limited Effort: this service is intended for dealing with the lowest priority traffic in a network, which only receives resources when it is not competing with any other service. This type of service has also been called Lower Effort [29], Bulk Handling, or Scavenger service [181].
- Class 8 - Signaling Handling: this service is particularly useful for transmitting protocol signaling information with a lower packet loss and delay than the best effort can provide.

In the next sections, a very simple example of a WDS Class and a WDS instancing is presented here, in order to shed some light on these two concepts.

4.4.2 WDS Class - Conversational Multimedia Service

The definition of this WDS class is given below.

- Service Identification

  WDS Class name: Conversational Multimedia Service (CMS)

  WDSID: 000001 (decimal 1)

- Service Description

  This WDS Class specifies a quantitative service intended to provide strict performance guarantees for real-time user services, which typically generate bidirectional traffic. It is a predictive service, since its QoS parameter bounds (delay, jitter, loss) are defined in terms of percentiles, and not rigid upper bounds.

- Service Level Specification

  - Scope: only the pipe style is supported by CMS as (*source domain: destination domain*). Source and destination domains are identified by their IP network address in the form *address/prefix-length*.
  - Schedule: forever (default).
  - Reliability: The contracted service must be available 99,9% of the time. In case of an outage, the service must be operational again in at most 2 hours.
  - Monitoring: End-to-end values of delay, jitter, packet loss and throughput are to be collected at 60-minute intervals by the Simple Interdomain Monitoring

Protocol (SIMP)[21]. Domains must report the average of measurements taken at 5-minute intervals.

- SLO:
    - Traffic Identification: Source and destination domain addresses from section "scope" are additional information for traffic identification. Intermediate domains may use this information for disaggregating traffic belonging to this service in order to have a better control on traffic conditioning.
    - Traffic Profile: Transmission rate is set up for each direction according to the results of the negotiation. Each time this service is renegotiated the transmission rate may be changed. The allowed burst size is the maximum of 1% of the transmission rate or the MTU.
    - Excess Treatment: Traffic submitted in excess must be dropped.
    - Performance Guarantees:
        - ❑ Delay: Refers to the one-way-delay, as defined by IPPM [7]. Two levels of delay are defined, representing both "good" and "acceptable" quality, each one associated with a time interval and a quantile.
        - ❑ Jitter: It is defined as the difference in the one-way-delay of two consecutive packets [60]. Jitter is represented as single value, associated with a time interval and a quantile.
        - ❑ Loss probability: It is the ratio of lost (in-profile) packets between the ingress router at source domain and the egress router at destination domain and the offered (in-profile) packets at ingress router. Loss probability is represented as a single value and a time interval.
        - ❑ Throughput: It is defined as the amount of traffic leaving the egress router at destination domain, i.e., the transmission rate of section "traffic profile", without the lost packets.
    - Direction: two SLOs must be instantiated, from source to destination and vice-versa.

[21] As this service is only an example, it is here assumed that this protocol does really exist.

- Formal Proof

  Not applicable for this WDS Class.

- Attributes and Constraints

  CMS is a bidirectional service and as such, some considerations are necessary:

  - Throughput may be different for the two directions. This may occur when, for instance, a user sends and receives information in different bit rates (due to asymmetric access networks such as ADSL and Bluetooth).
  - The negotiation model used by the CDG must allow bidirectional negotiations and resource reservations.

- Examples of Use

  This WDS class may be used to deploy end-user services such as telephony, videophone and videoconferencing.

- Guidelines for Implementation

  Domains must take care with routing asymmetry when implementing CMS. Since the service is to be negotiated as a whole, CMS is applicable in situations where routing is symmetric by its very nature or when the negotiation process operates in the routing active style (thus being able to change routing tables).

4.4.3 WDS Instance

In order to perform the instantiation of a WDS class, only the missing parameters are needed. This WDS Instance is a permanent one, since it has its own WDSID.

- WDS instance

  WDSID: 100001

  WDS Class ID: 000001

- Service Level Specification

  - SLO 1:

    - ■ Performance Guarantees:

      - ❑ Delay:

            - Level 1 (good): 70 ms, 99,9%, 60 minutes.
            - Level 2 (acceptable): 320 ms, 99,9%, 60 minutes.
        - Jitter: 20 ms, 99,9%, 60 minutes.
        - Loss probability: 10^{-4}, 60 minutes.
    - Direction: source domain to destination domain.
- SLO 2: (it is a repetition of SLO 1, inverting the direction)
    - Performance Guarantees:
        - Delay:
            - Level 1 (good): 70 ms, 99,9%, 60 minutes.
            - Level 2 (acceptable): 320 ms, 99,9%, 60 minutes.
        - Jitter: 20 ms, 99,9%, 60 minutes.
        - Loss probability: 10^{-4}, 60 minutes.
    - Direction: destination domain to source domain.

4.5 Service Identification Scenarios

One of the main issues related to WDS implementation is how to identify packets belonging to the traffic stream of a WDS in order to provide them the required performance guarantees. Within a CDG, the ultimate end-to-end identification of a WDS is through its WDSID. The DS field (Differentiated Services Field) is used in Chameleon for implementing the WDSID by identifying every packet belonging to a given WDS. This scheme will not interfere on the normal use of the DS Field for selecting PHBs inside DS domains, since it will be used as the WDSID in interdomain links. As it is a 6-bit field, it allows up to 64 different WDS classes or instances[22]. If parsimoniously used, the maximum number of simultaneous WDSs will not be exceeded. The reasons for believing this to be the case are twofold. Firstly, it is expected that only a small number of WDSs will be enough for deploying a great deal of end-user services. Secondly, when interconnecting services from two or more CDGs, each one may

[22] Generally, only WDS classes have their own WDSIDs. However, for some highly used WDS instances, the Service Maintainer may create a permanent instance and assign to it a WDSID. This operation is aimed to make service negotiation simpler and easier, although it must be done with care, in order for the CDG Controller not to run out of available WDSIDs.

use a different WDSID. Hence, a particular WDSID does not need to have a global meaning in the whole Internet, although this is highly recommended for the sake of simplicity.

The WDSID does not need to be used by every two neighboring domains, though. Some domains may choose to use a local identifier (LSID – Local Service Identifier), agreed upon in some phase of the negotiation process (a bilateral agreement). When traversing interdomain links, packets are identified either by the WDSID or the LSID. For example, the LSID may be the DiffServ Code Point (DSCP) for two domains that implement a certain WDS using the same technology (DiffServ) and the same PHB. A whole domain group may do it when they belong to the same DS Region.

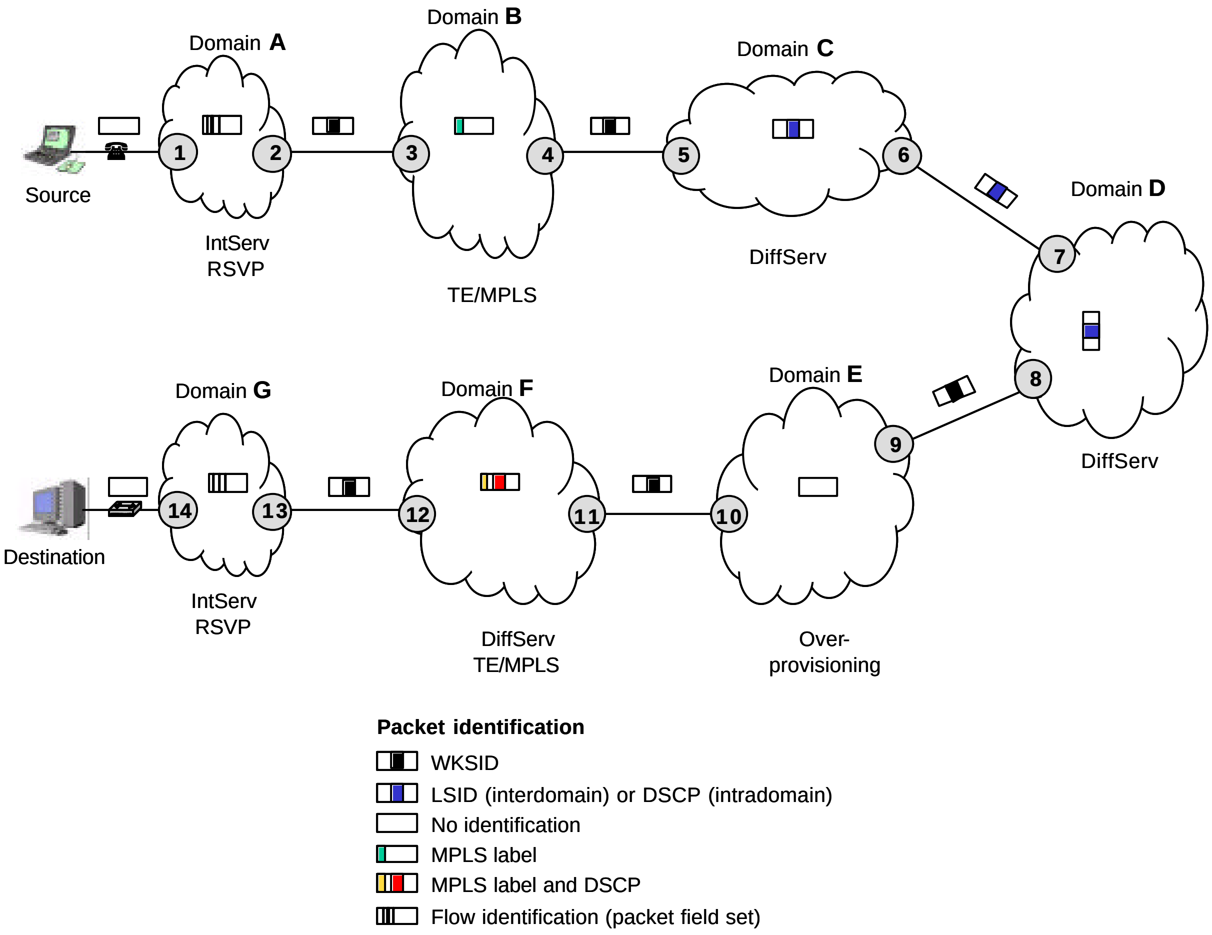


Figure 4.1 - Packet identification in an end-to-end service

Figure 4.1 shows how a sample packet from an end-to-end service may be identified inside and outside domains as it gets forwarded from source to destination, traversing domains A through G. In each domain, the ingress and egress routers are represented as numbered circles. The actions taken in each of the border routers with regard to the policies for packet identification are as follows:

(1) Domain A's ingress router: As domain A uses IntServ/RSVP, it is assumed that an RSVP session was previously setup from source host to egress router 2. Router 2 plays the role of an RSVP proxy (section 3.3.4). At router 1, the packet is identified by some of its layer 3 and 4 fields (according to IntServ classification rules) and receives the suitable forwarding treatment.

(2) Domain A's egress router: The sample packet is market with the WDSID in its DS field, according to marking policies coordinated by the service/operation mapping functional module (section 3.2.1) of the service plane.

(3) Domain B's ingress router: Router 3 maps the WDSID to an LSP (MPLS path) able to provide packets the required QoS guarantees and inserts the corresponding label in the packet, leaving the WDSID unchanged. The mapping policies could even be more sophisticated if domain B divides up traffic of a WDS into different aggregations and forwards them through different LSPs. In such a scenario, traffic identification information from the SLS could be used together with the WDSID.

(4) Domain B's egress router: No action is needed here, since the packet maintains the original WDSID. The MPLS label is normally removed.

(5) Domain C's ingress router: Domain C is a DiffServ domain, and the internal DSCP (used for selecting the appropriate PHB) is different from the WDSID. Hence, the packet must be remarked at router 5.

(6) Domain C's egress router: No action is needed, since domains C and D had previously agreed upon a common LSID, which is the same value as the internal DSCP used by both domains.

(7) Domain D's ingress router: No action is needed, since the LSID and the internal DSCP are the same.

(8) Domain D's egress router: In router 8, the packet must be remarked again to the WDSID. The mapping policy might be based only on the LSID or on other fields, in case more WDSs had been mapped to the same DSCP.

(9) Domain E's ingress router: No action is needed, since domain E relies on over-provisioning instead of using a QoS technology.

(10) Domain E's egress router: No action is needed, since the packet maintains the WDSID.

(11) Domain F's ingress router: Domain F is a DiffServ/MPLS domain and as such, the PHB may be determined by the DSCP (e.g., ingress and egress router) or by the MPLS label (e.g., inner LSRs). The detailed operation depends on domain F's particular choice for DiffServ operation over MPLS [137]. Therefore, the original WDSID must be mapped to both an internal DSCP and an MPLS label.

(12) Domain F's egress router: The packet must be remarked again to the WDSID.

(13) Domain G's ingress router: Domain G uses IntServ/RSVP. Similarly to what happened in domain A, it is assumed that an RSVP session was previously setup from ingress router 13 to the destination host. The packet is identified by previously configured IntServ rules, and no specific actions are needed.

(14) Domain G's egress router: The same reasoning of router 13. The packet is finally forwarded to the destination host.

An interesting point is where the mapping between end-user services and transport services should be made. The effects of a WDS are within its defined scope (section 4.3.1), which might not cover access networks and certainly does not cover customer premises. For example, it is hard to provide users uniform guarantees as they may be connected to different access technologies, such as dial-up lines, ISDN, ADSL, cable modem or GPRS (section 3.3.4). For this same reason, performance guarantees in customer shared Ethernet networks are outside the responsibility of access providers. In any case, establishing where the mapping should happen is left to the user service provider discretion. For example, if the end-user application sets up an RSVP session before starting to send user data, the mapping effectively happens at the source host, even though WDS performance guarantees are applied to the traffic only at the access provider first-hop router. The end-user application could optionally also mark packets with the WDSID or with an internal DSCP.

A second related point is where the packet should be marked with the WDSID for the first time (other than in the source host). In this example domain A's egress router (router 2) is the responsible for this marking operation, since it is the RSVP proxy and it must determine which level of treatment the packet must be given by downstream routers. The first-hop router (router 1) could also perform this marking operation. However, concentrating this operation only in the egress router is much more efficient, as there is no benefit in having marked packets inside the IntServ domain's A network. Alternatively, downstream domain's ingress router (router 3) could mark packets on behalf of domain A if this service was part of their particular bilateral agreement.

This example does not cover every possible service identification scenario, but provides a good understanding on how this operation could be done with the known existing QoS implementation approaches. Yet another important aspect is that in Figure 4.1 were considered seven different domains, which is not true in most cases in today's Internet. Traffic flowing from any source to any destination generally crosses between 3 and 6 different domains [160].

4.6 Related Work

WDS is not a complete brand new concept, but it has been evolved from recent developments in the area of QoS in the Internet. The most distinctive feature of Chameleon's WDS idea is that it tries to define end-to-end quality of service levels without relying on a particular QoS implementation technology (such as IntServ or DiffServ). Using a single technology makes it easier to implement an end-to-end service. However, given the current heterogeneity of the Internet, it is not reasonable to assume that the whole Internet will agree on a single QoS technology. This may be true in highly controlled corporate networks or in small parts of the Internet, though.

IntServ aimed to extend the traditional Internet's best effort service by defining new service classes with QoS guarantees. Although the definition of service adopted by IntServ refers to a "set of QoS control capabilities provided by a single network element" [183], the body of any service specification document must include an "end-to-end behavior" section. By concatenating the effects of a series of routers implementing an IntServ service, a true end-to-end service can be built. Chameleon's WDS incorporates the IntServ idea of precisely defining an end-to-end service behaviour in order to deploy advanced services in the Internet. However, in addition to the well-known scalability problem (section 2.3.1), the IntServ approach for defining services raises other concerns. Firstly, the development of new services is not flexible, because IntServ requires services to be standardized. The IntServ work group only standardized two services, the "guaranteed quality of service" and the "controlled load service", whereas in Chameleon any CDG is free for deploying whatever services the domains see fit. Secondly, IntServ obviously requires the same QoS technology to be implemented in every router and domain in the whole path from source to destination.

In DiffServ, the approach is defining Per-Hop Behaviours (PHB), which specify different forwarding treatments. The idea is that end-to-end services may be built on top of these basic functional components. Each service relies on a particular PHB and traffic conditioning mechanisms. This task is not straightforward, though, because turning forwarding treatments into end-to-end services requires careful domain provisioning and configuration, along with negotiation between neighboring domains [25]. In order to help domains in deploying DiffServ-based services, the working group came up with the concept of Per-Domain Behaviour (PDB), which may be though as an edge-to-edge service. DiffServ is a suitable technology for providing service differentiation with QoS guarantees inside domains, but neither the PHB nor the PDB are sufficient concepts for deploying end-to-end services, hence remaining an open question.

There have been some attempts of using DiffServ as a basic QoS technology for defining end-to-end services, such as TEQUILA, AQUILA and Internet2 QBone's Bandwidth Broker. In TEQUILA, the strategy is to standardize a SLS format in order enable possible service definition and negotiation between service providers and their customers as well as between peer providers in the Internet. Tequila's SLS allows the negotiation of ad-hoc services, in contrast to Chameleon, where WDSs must be pre-defined for any CDG (section 4.1.1). Furthermore, TEQUILA considers DiffServ as its basic QoS technology, although allowing IntServ in access domains and MPLS for traffic engineering.

AQUILA went beyond this generic SLS description of TEQUILA and introduced the concept of Predefined SLS Types, in order to simplify service deployment. This concept has emerged in parallel with Chameleon's WDS and they have some similarities and differences. A Predefined SLS Type has functionalities of both WDS classes and instances (section 4.3). It fixes SLS parameters and particular values for them at the same time. Chameleon' WDS allows more flexibility in service definition, by making it possible to introduce slightly different services to be deployed by a CDG simply by instantiating a WDS class with different parameters. Furthermore, in Chameleon the SLS is only a part of a WDS, which gives a more precise definition of the intended service than an SLS is able to.

The Global Well-Known Service (GWS) concept defined by the Internet2 QBone project, was the stronger influence on the development of Chameleon's WDS. However, the concept of GWS is very loose (not well developed) and it only has been conceived for defining the QBone

Premium Service (QPS)[23] [192]. As a GWS, the QPS has and identifier (GWSID) and needs to be globally understood by every QBone member. The goal is maintaining the end-to-end service semantics and simplifying the negotiation by Bandwidth Brokers through the SIBSS protocol [192]. In contrast, Chameleon's WDS gives a more precise definition of a service and is also more flexible as it must be known only within a CDG.

In CADENUS, a service may be seen as both an user service and a transport service, depending on the context. Every CADENUS service is described by an SLA, which may be a retail SLA or a wholesale SLA, for user and transport services, respectively. Services with similar semantics are described by SLA templates [185] (similar to WDS classes), from which SLA instances may be created. However, there is no explicit attempt to identify services for simplifying end-to-end service negotiation. Furthermore, a SLA may be also a composite service, which is a recursive reference for a more complex service comprised of other services. SLA templates can be designed for satisfying customer needs, in contrast with the goal of keeping small the number of WDSs in Chameleon. CADENUS uses the SLS format developed by both TEQUILA and AQUILA projects.

In GÉANT, an IP Premium Service [36] has been designed and implemented [173] for providing users a service equivalent of a leased line. GEANT service specifications are based on the above-mentioned projects. The IP Premium Service is a derivative of AQUILA's predefined service that uses an approach based on CADENUS' SLA templates and a TEQUILA'SLS format [37]. A distinctive feature is that it allows the definition of bidirectional services, by configuring SLOs in two opposite directions [179].

4.7 Summary

This chapter presented an in-depth view of service definition in Chameleon, which is based on Well-Defined Services. A WDS combines some very useful features that are only partially covered (or not at all) by other approaches:

- A WDS gives a precise definition of service semantics and it is not tied to a particular QoS technology. Packets can be easily identified as belonging to a particular WDS and given the right treatment in order to preserve the service end-to-end semantics.

[23] Since the deployment of QPS has not been successful [196], the Internet2 QoS Working Group is now focusing on a qualitative service called QBone Scavenger Service (QBSS) [181].

- WDSs provide flexibility in creating service classes and instances. Although domains do not have freedom for negotiating ad-hoc services (as in Tequila's SLS), they can create WDS instances with different parameters.
- It is aimed at facilitating service negotiation by a large number of domains simultaneously. The focus in TEQUILA and AQUILA, for instance, is the negotiation between customer and provider.
- It permits service negotiation as a whole, including bidirectional traffic. Other approaches have been considering a service only for unidirectional traffic. Bidirectional traffic needs two SLSs, one for each direction.

In the next chapters, the important function of service negotiation is presented and evaluated. Once services are well defined by WDSs, the negotiation process becomes easier to tackle and deploy.

Chapter 5

Dynamic Interdomain Service Negotiation

Interdomain service negotiation refers to the process whereby domains communicate with each other in order to deploy services in the Internet. It involves verifying the feasibility of deploying a given service through some path (sequence of domains) in terms of its required performance guarantees. In the current Internet, this negotiation is performed in a static manner, for providing simple connectivity to the best effort service. Domains maintain several bilateral agreements in order to exchange traffic and routing information with each other. The timescale for renegotiation typically is in the terms of months, depending on the evolution of the observed traffic. With the introduction of new advanced services, this situation tends to become more critical, due to the more frequent changes on service utilization patterns and the urge to finding new routes for meeting services' performance requirements. Therefore, *dynamic* service negotiation will be necessary, and should be based upon efficient, scalable, fair and financially viable negotiation models.

This chapter presents two of the most important contributions of this thesis in the context of the Chameleon architecture, namely approach for dealing with the negotiation of transport services and also the proposal of the hierarchical negotiation model. Section 5.1 presents an overview of the service negotiation in Chameleon, as being comprised of user and transport service negotiation. Section 5.2 elaborates further the rationale for the above-mentioned need for dynamic negotiation. The negotiation process in Chameleon, comprised of SLA, WDS and resource negotiation is described in section 5.3. In section 5.4, a classification based on styles is proposed, that will be used further for characterizing the negotiation models. Five negotiation models, the cascade, hub, hierarchical, wave and border models are presented in sections 5.5, 5.6, 5.7, 5.8 and 5.9, respectively. Section 5.10 addresses the aspects of negotiation among different CDGs, for both situations where homogeneous and heterogeneous negotiation models are being used. Related work is presented in section 5.11. Section 5.12 summarizes this chapter by highlighting its contributions.

5.1 Service Negotiation in Chameleon

Service negotiation is an important part of the whole service life cycle, which involves service definition, negotiation, provisioning, creation, utilization, monitoring, authentication, authorization, accounting, pricing and billing. In Chameleon, service negotiation is a process made of two different phases: user service negotiation and transport service negotiation. They are distinct and primarily intended to proceed asynchronously. In other words, transport service negotiation should not be triggered by a user service negotiation (see discussion on periodic and urgent negotiations in section 5.4). The organization of the service negotiation dynamics in two phases results from the Chameleon's service deployment model.

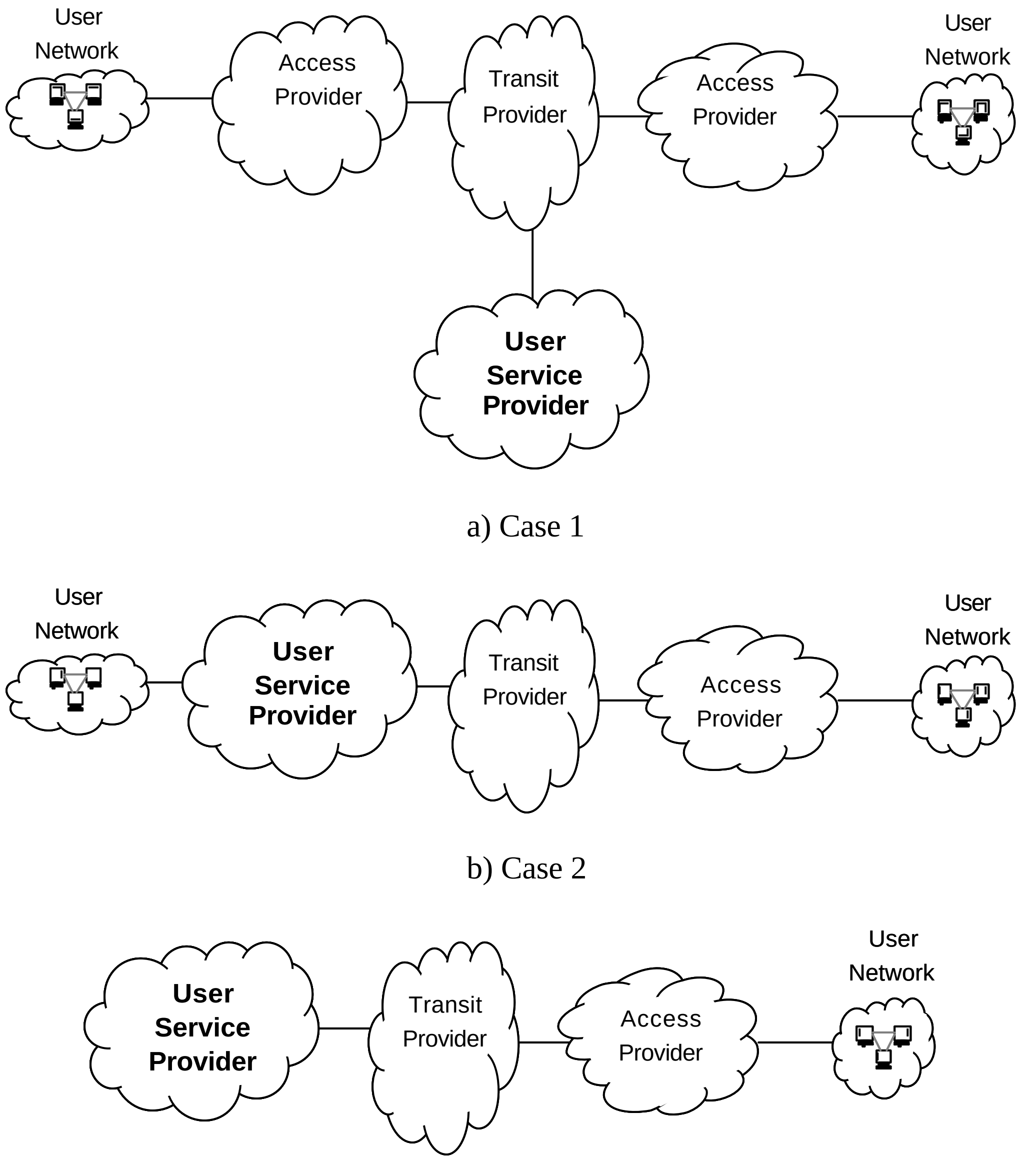
5.1.1 Service Deployment Model

It is the user's current point of view that a service refers to access to the Internet, whereas quality of service is associated to access link capacity (and technology). In such a scenario, it is natural for users to think that their access provider is the service provider. As the Internet evolves from a single service model (best effort service), whereby quality and pricing are based on bandwidth, to a true multi-service network, there should be an increase in the growth of the number and diversity of User Service Providers (USP). The user (assuming that it is the customer) will negotiate directly with the USP, which in turn will negotiate with other domains, in order to set-up the network infrastructure for the correct service operation.

The rationale behind this model is that having physical access to the Internet will gradually become less important than having permission to access advanced services. In this context, USPs will assume the functions of keeping user information and also dealing with customer relation management matters. Physical access will probably become a commodity and USPs will broaden their reachability by having contracts with as many Access Providers (APs) as they are able to have.

At this point, there are still some missing pieces for completing the service deployment puzzle, in order for QoS-based services to provide end-to-end performance guarantees. Instead of establishing bilateral agreements with many APs and having to deal with end-to-end service guarantees, USPs may also be members of Chameleon Domain Groups (CDGs). Thus, they can make use of the infrastructure provided by the CDG for deploying WDSs and automatically have their services used in as many different locations as the CDG's geographical scope permits (a CDG scope may be extended by interconnecting to another CDG – section 5.10).

The USP may be topologically located in different places, with respect to the traffic (service) path and the direct (routed) path between user networks. Figure 5.1 depicts three possible cases:



a) Case 1

b) Case 2

c) Case 3

Figure 5.1 – Possibilities for the topological location of User Service Providers

- Case 1: the USP is not on the direct path between user networks. There are two variations: a) users communicate directly with each other, as in a telephone IP/IP service (host to host); b) communication is through the USP, such as in a game service with a centralized server.
- Case 2: It is a simplification of case 1, whereby the USP is on the direct path between user networks. In this case, the USP is also the AP.
- Case 3: the USP is the data source or destination, as in a Video over Demand (VoD) service, where the USP streams the video to the user. Another example may be a telephone IP/PSTN server, whereby the USP is the gateway to the PSTN.

5.1.2 User Service Negotiation

The big picture of advanced service utilization starts with the user, which must invoke an USP and request a given service (e.g., a voice or video service). The USP accepts or rejects the request, after checking whether it is able to provide this service within the requested scope. This process is called the user service negotiation, which may be done by whatever means the USP is able to offer. For instance, the user may fill in a form on a web page, giving enough information (including billing information). Whether or not the USP provides a formal user SLA is a business matter and involves only the user and the USP. User service negotiation does not produce any kind of resource provisioning, only the intention of both users and providers in purchasing/selling a service.

A possible scenario for the user service negotiation and its resulting developments are presented in Figure 5.2. It is assumed that the USP and the AP have a previous multilateral agreement with a CDG. This scenario is comprised of 6 phases:

1. User service request: The user requests a game service by a particular mechanisms offered by the USP. The USP maintains a game server (a highly common service in today’s Internet) but also provides performance guarantees for its users.
2. User service accept/reject: The request is accepted or rejected, as a result of the USP’s analysis. There can be several reasons for a service being rejected including business ones (e.g. problems with the user’s credit card). A very simple technical reason could be the lack of service availability within the requested scope. In this scenario, the USP has sufficient information to allow it to decide whether an additional user may access

this service from that particular AP. A different scenario might include another phase so that the USP could inquire the AP about service availability. If the service is accepted, the user may be given a time limit for starting to effectively use the service.

3. User service registration: The USP registers and accounts an additional user for the game service.

4. User service notification: This phase involves the agreement between the USP and the AP. The USP notifies the AP about the new users (it may wait a while and aggregate more users in the notification message) who are authorized to use that service.

5. User service to transport service mapping: This phase consists of configuring the mapping from user services into transport services for each user, according to section 4.3.2.

6. User service utilization: The user makes actual use of the game service. An improved scenario may also require the use of stricter authentication and admission control mechanisms.

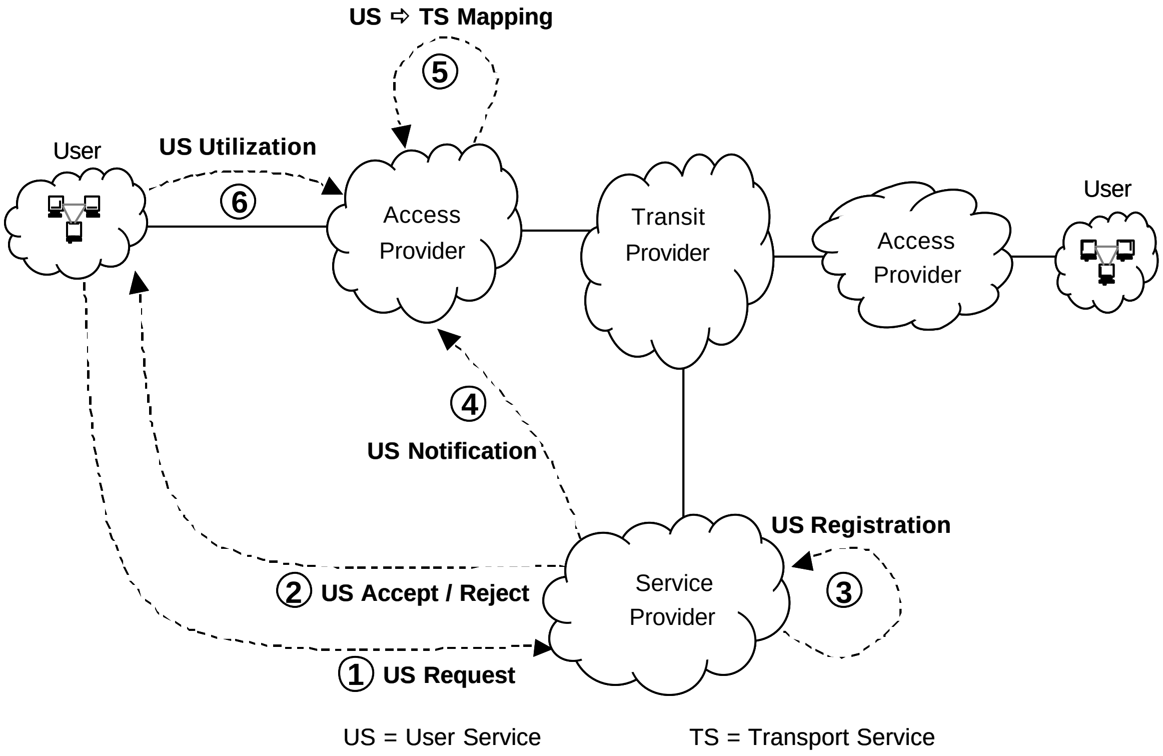


Figure 5.2 – User Service Negotiation Dynamics

This scenario is an instance of Figure 5.1.a, where the USP forwards the messages among users. Should the USP be located in the direct data path (Figure 5.1.b), phases 4 and possibly 5 would not be necessary.

The actual user service negotiation process is out of the scope of Chameleon. The same is with the negotiation between USPs and APs. Chameleon is focused on transport service definition and negotiation and throughout this thesis it is assumed that user service configuration, negotiation and utilization may be performed by other means. User service negotiation provides an input for transport service negotiation, but they are orthogonal activities. The CADENUS Access Mediator and Service Mediator functionalities [164] are good candidates for implementing user facilities.

5.1.3 Transport Service Negotiation

The scope of user service negotiation is the edges of the Internet: users, APs and USPs. On the other hand, transport service negotiation makes the bridge among the edges by providing performance guarantees for user services through the Internet core. In Chameleon, the interdomain service negotiation must be based on Well-Defined Services (WDSs). This approach makes service negotiation simpler and easier, by limiting parameter and value combinations (section 4.1.1). In the next sections, service negotiation refers to interdomain dynamic transport (WDS) service negotiation.

Domains may play the roles of service buyers or service sellers. For an end-to-end negotiation style, USPs are always service buyers, whereas Transit Providers (TPs) are always service sellers, unless they are also APs, where they may play both roles.

- Service Buyers: Domains that are either the source or the destination of user service traffic. As such, they are responsible for requesting services and paying for them. They have commercial interest in finding out whether or not a given service can be deployed over a given scope and purchasing enough resources for providing the end-to-end performance guarantees that user service requirements. Service buyers should be able to do traffic measurement and prediction in order to have reasonable estimates about the future traffic. Either USPs or APs may assume the function of service buyer during service negotiation. The USP will most likely be the service buyer in situations where it is the source or destination of user service traffic. When the USP is not able to play the role of service buyer, it must rely on the user AP, which must in this case have some financial compensation. The actual roles played by each domain depend on the business model of each CDG. Service buyers must provide values for purchase parameters (section 4.3.2) when requesting a service, including the service scope, for example.

- Service Sellers: Domains that have configured their networks for implementing certain WDSs and are willing to offer them to other domains. Service sellers may be either APs or TPs. When the user service involves two (or more) APs, one of them is a buyer and the other is a seller. The service seller must provide performance values for sale parameters (section 4.3.2) when offering a service, for instance throughput and delay.

From a service buyer perspective, the negotiated outcome for each purchased WDS may be:

- Service total grant: The service buyer was granted everything it requested.
- Service partial grant: Not enough resources were found in the end-to-end path for serving the request, in such a way that the service buyer was granted only a fraction of the requested resources (a percentage).
- Service denial: There are different reasons for a service request to be denied, these includes: a) The WDS cannot be deployed within the requested scope because one or more of the domains do not implement that WDS; b) one or more definition parameters, such as delay or jitter (section 4.3.2), are violated in the end-to-end path; c) no resources were found in the end-to-end path for that WDS or the amount of resources is below the minimum requested; d) a given domain does not want to offer that service for the requesting domain, due to internal domain policies (this reason may be masqueraded as something else, because domains may not want to reveal their policies).

These three possible outcomes may be classified in a single spectrum of “service grant”, where the level of grant may vary from 0% to 100%. Service grant implies that resources will be provisioned for that service all the way along the path from source to destination domains. While the goal is to purchase and sell services, the way the negotiation is done in a given CDG depends on the adopted negotiation model.

5.2 The Need for Dynamic Service Negotiation

Service negotiation[24] may be static or dynamic. It is considered static when it is negotiated by human agents, using any available means, such as a document, a phone call, an email or a live conversation. This is the way that interdomain link capacity is currently negotiated in the Internet. Despite its simplicity, static negotiation imposes an unnecessary lack of flexibility to the negotiation process. Generally, a provisioned interdomain link capacity cannot keep up with increasingly growing traffic rates, resulting in a often congested situation mainly at private peering points and NAPs[25] [3][5][105][113]. With the changes in the Internet infrastructure, and interdomain relationship requirements that will be necessary for deploying advanced QoS-based services, static negotiation will perform poorly, thus restricting the dissemination of such services.

In order to better match the requirements of real-world operational scenarios, domains will be induced to renegotiate their contracts periodically, in a dynamic fashion. Dynamic service negotiation precludes the intervention of human agents, allowing an automatic exchange of service purchase and sale information. Some contributing factors for the need of dynamic service negotiation are:

- Traffic volume changes: In any point of the Internet, traffic volume changes considerably over time, even when only considering the current single best effort service. The deployment of new services will introduce traffic with different characteristics and will disaggregate traffic of the best effort service into two or more new services. Other sources of traffic volume changes are:
    - Application characteristics [185]: Different applications generate different traffic patterns and volumes. Additionally, the utilization profile of specific applications varies greatly.
    - Time influence: A well-known characteristics of both voice traffic and data traffic is that they present significant variations in different time-scales, such as hour, day, week, month. Some on-line statistics, to confirm this, may be found in [2] and [171].

[24] From this point on, service negotiation refers to interdomain transport service negotiation, unless something different is stated.

[25] As a matter of fact, there are other reasons that also contribute to the congestion in peering points and NAPs, such as a lack of financial incentives (sections 2.1.7 and 6.7.2) and political matters, but here the focus is on technical reasons.

  - Subscription variation: The number of Internet users is growing continuously [197]. As advanced services are deployed in the Internet, probably users will have an increasing interest in subscribing to them. At some point in time, some services will become very popular and next they may become less attractive, when users will focus on other new services. Therefore, traffic generated by these services may grow and shrink. Such a situation happened recently with NAPSTER.
  - User mobility: The predicted explosion of mobile Internet users in the near future [209] and the advent of overlay networks providing vertical hand-offs [75][188] between different wireless technologies, will allow data sources and destinations (users) to quickly migrate from one service provider to another or between different regions of a given service provider.

- Settlement schemes: The standpoint in Chameleon is that the deployment of advanced services requires settlement schemes based on both resource reservation and utilization. In this scenario, domains will not request (during negotiations) much more resources than they expect their users will effectively use. On the other hand, they will try to have at any one time enough resources for coping with users' demands. As traffic varies over time, this tends to influence domains decision for renegotiating services more frequently.
- Resource utilization: Domains always try to optimize their network resource utilization. By allowing service negotiation to be dynamic, domains will have the opportunity of engineering their networks for a better utilization of resources [177][179].
- Service offerings and pricing strategies: Either for technical or for business reasons, some domains of a given CDG may opt for not deploying some services or for only deploying them during time periods. They are free to change their policies and present different services offerings in further negotiations. Similarly, domains may use a variety of pricing strategies, e.g., offering special discount for hours or days with a light network load [25].

5.3 Service Negotiation Process

The service negotiation process is divided up into three phases: 1) SLA negotiation; 2) WDS negotiation; 3) Resource negotiation. Each of which has different objectives and renegotiation timescales. The goal of having these three phases is organizing and simplifying service negotiation, in order to allow more effective service management. Phases 2 and 3 may be done simultaneously, depending upon specific agreements of CDG members and technical reasons, such as the negotiation model (which may not allow phases 2 and 3 to be executed separately).

5.3.1 SLA Negotiation

The first phase of the service negotiation process is a high-level one, which involves commercial and legal clauses, and it is negotiated by human agents. The SLA negotiation is a kind of umbrella contract, which permits further specific WDS and resource negotiations. The typical timescale for renegotiations is in terms of months or years. Additionally, domains must negotiate a SLA when becoming new members of a CDG. The CDG Controller has the responsibility for the SLA management.

All CDG members have to sign a common SLA (section 4.2), which is in essence a multilateral agreement. However, this SLA is different from a typical existing multilateral peering agreement of a NAP [43][158], whereby a domain peers with every other domain. As stated in section 3.1.2, CDG members are not expected to be fully interconnected. Furthermore, Chameleon provides domains with financial incentives for participating in a CDG, by discriminating exchanged traffic according to WDSs and making it possible to charge on a request and/or usage basis (section 6.7.2). Therefore, the argument that multilateral agreements rarely work [151] may be overcome with the deployment of the Chameleon architecture.

Technically, the SLA negotiation phase offers the benefit of deciding about service definition and negotiation. In this phase, CDG members decide: a) which WDSs (both WDS classes and permanent instances) will be deployed by the CDG; b) whether these services are well known (WKS) or local (LS) ones; c) the negotiation model to be used by the WDS and the resource negotiations phases. Other arrangements can be made, that will have some influence on the other two negotiation phases. For example, a special Signaling Handling WDS may be statically instantiated (section 4.4.1). It is aimed at carrying traffic of WDS and resource negotiation messages and possibly of other QoS-related operation signaling protocols, such as SIP, H.323, RSVP, LDP, RCTP, as well as any other new or existing protocols (e.g. those protocols identified by the IETF’s NSIS working group).

Domains may also make use of the SLA Negotiation phase for making specific bilateral arrangements with peering domains. These arrangements are exclusively intended to resolve bilateral problems or improve performance and must not interfere in the normal service operation. For example, a stub (user) and a transit domain may negotiate a specific QoS technique (e.g. RSVP) that will be used in the operation plane for some services. Another example is the negotiation of a local identifier for the WDS (LSID) that will be used instead of the WDSID defined by two peer domains.

5.3.2 WDS Negotiation

WDS performance parameters may be of two types (section 4.3.2): definition parameters (delay, jitter and packet loss) and negotiation parameters (throughput). Even though there is a need for frequent and dynamic service negotiations, domains are expected to forward traffic keeping definition parameters within certain bounds. This is especially important for stricter services, as they may exert higher control over their networks, by means of QoS technologies. The quality of the deployed services also depends on the stability of QoS levels that the network is able to offer. However, throughput is the parameter that domains are likely to renegotiate more frequently, according to instantaneous traffic variations. Hence, it is more suitable not to renegotiate definition parameters (which may involve complex interactions or calculations, depending on the negotiation model) every time throughput is renegotiated. A typical timescale for WDS negotiation is in terms of days or weeks.

The WDS negotiation phase is a dynamic one, performed by software agents, i.e., the Service Brokers (SBs). Its goal is verifying whether the requirements of the definition parameters can be met in an end-to-end basis. The implementation of the WDS negotiation is highly dependent on the negotiation model. The WDS Negotiation phase may occur together with the Resource Negotiation phase, depending on particular features of each WDS, on the negotiation model and on CDG decisions. The idea behind having two conceptually separated phases is that it is very useful for understanding the implications of different WDS performance parameters (definition and negotiation parameters) in the negotiation.

The WDS negotiation is focused on WDS instances, which may be either permanent ones (defined in the SLA negotiation phase) or on-demand ones. There are at least two reasons for starting a WDS negotiation:

- For the whole CDG: automatic negotiation for checking if WDS instances may be deployed through all domains wishing to do so.
- At the discretion of domains: domains initiate the WDS negotiation to verify whether a given WDS may be deployed in a given scope.

The results of the WDS negotiation phase, as seen by a CDG member, may be threefold:

- A $M \times S$ matrix of authorized services, where M is the number of destination domains and S the number of services. Each cell contains the status ("accepted" or "rejected") of the WDS negotiation for one service to one destination. For each accepted service, it results in the information of the negotiated value or level of each definition performance parameter. In case of the service being rejected, there is a rejection code is provided. It is worth mentioning that this phase does not allocate resources, that is, throughput is not negotiated. Therefore, even though a service is accepted, it is still necessary to obtain the associated resources in further negotiations.
- A commitment that domains are able to forward traffic while respecting certain bounds. The end-to-end negotiated value for a parameter is a combination of the capabilities of each domain. Delay and jitter are additive parameters, whereas packet loss is multiplicative [215]. For example, lets suppose a service S1 is accepted from domain A to domain Z with two transit domains between them, T and U. A 70 ms delay was negotiated, of which both domains A and Z account for 10 ms, domain T for 20 ms and domain U for 30 ms. Each domain makes a commitment to forward traffic within this maximum delay.
- Interdomain routing changes. In the event that optimization is supported by the negotiation model (when it allows the routing active style - section 5.4.6), the WDS negotiation could be enabled to change the BGP routing tables.

5.3.3 Resource Negotiation

Using the WDS negotiation phase, domains become aware of which services can be deployed extending across certain scopes (pairs of source and destination domains). The next phase of the service negotiation process is to check if there are enough resources for those services in the end-to-end path. Resource negotiation always refers to WDS instances.

The following important concepts related to service negotiation are introduced next:

- Resource: throughput is the item to be negotiated, represented in bits per second (bps) units.
- Resource request: buyer domains request a given amount of resources (for services) from other domains, based on the resource request matrix (section 3.5.1).
- Resource offering: seller domains offer a given amount of resources, based on the service offering matrix (section 3.5.2).
- Resource allocation: process of looking for resources for current requests, while trying to optimize the use of available network resources.
- Resource grant: requested resources may be granted totally or partially. The amount of resources granted is the lowest value available in every domain and interdomain link in the path. Granted resources are represented by their percentage with relation to the requested resources.

A resource grant implies that resources will be provisioned for that service all the way in the path from source to destination domains. The way that domains may interact for performing resource requests, offerings, allocations and grants may vary substantially, depending on the particular negotiation model chosen by the CDG.

The typical expected timescale for resource renegotiation is shorter than SLA negotiation and WDS negotiation, in terms of hours or days. The CDG may agree on renegotiation periods of one hour if domains foresee a very dynamic scenario. As traffic varies considerably during a day, domains may need to perform multiple negotiations in advance, for multiple intervals or hours (24 one-hour intervals, for instance).

5.4 Service Negotiation Models and Styles

The service negotiation model determines the particular mode whereby domains in a CDG interact in order to achieve automatic WDS and resource negotiations. Only the service plane is aware of service negotiation in Chameleon (more specifically, the Service Broker). Since it is logically separated from the operation and monitoring planes, a CDG may choose to use initially a given negotiation model and exchange it for another one afterwards, without adversely impacting the service deployment, at least in theory.

A negotiation model presents itself as a logical and schematic view of the interaction among domains. As such, negotiation protocols intended to implement a negotiation model are not necessarily comprised in a negotiation model description. However, a description may give hints for helping protocol designers, for instance, messages that are expected to be exchanged between the involved entities.

The design of a negotiation model involves some decisions that must be taken, with regard to different possible styles, including scope, control, synchronization, timing, direction, and routing. In the remaining part of this section, a in-depth classification of negotiation styles is presented.

5.4.1 Negotiation Scope

Negotiation scope[26] refers to the extent that domains deploying an end-to-end service are involved in the automatic negotiation. So far, the negotiation scope has been considered an end-to-end scope, as stated in section 1.3. However, some negotiation scopes that are not end-to-end may also be used.

End-to-end scope: In a negotiation model based on an end-to-end scope, the negotiation extends across all involved communication ends. This means that the service availability was checked all the way from source to destination domains (and backwards, for bidirectional services), and once resources are granted, the buyer domain may rely on the performance guarantees provided for that service. In order for this style to work properly, domains need to know in advance their resource requirements to every other domain they are deploying end-to-end services together with. This implies that domains must be able to perform destination specific traffic measurements and prediction. Where this is not feasible or that the service scope is too broad (any-to-any) other styles of negotiation scopes may be more suitable.

Intermediate scope: An intermediate scope suggests that as a result of the negotiation, buyer domains are not given guarantees of end-to-end resource availability. This style of negotiation scope is easier to implement, but it needs to rely on other complementary means to obtain end-to-end QoS guarantees. An example may be using service negotiation only in some

[26] Negotiation scope should not be confused with service scope (section 4.3.1). The latter may be larger than the former.

parts of a CDG. Domains that are well known for keeping their networks over-provisioned may not participate in the service negotiation. This is not the envisioned scenario for the deployment of the Chameleon architecture, but it may be ultimately deployed by a CDG, at the discretion of the participating domains.

Border-to-border scope: Border-to-border may be considered as an extreme case of an intermediate scope, where negotiations involve only the next-hop domain for a given service. In its static form, it has been used for ages in the telephony world (PSTN). As every domain needs to keep its network working properly independently, all domains behave as service sellers and buyers. In such a scope, end-to-end guarantees come from the concatenation of multiple bilateral guarantees. Guarantees are based on provisioning, and in order to have these scaling to a true end-to-end scope, domains must rely on each others. This negotiation style assumes a well-provisioned network and fairly well known traffic patterns and volumes.

A border-to-border negotiation makes negotiation easier, but its applicability becomes narrower. The business model may be more complicated or it may limit the possible services. For instance, who should pay for a service? This is easier in an end-to-end negotiation (the service buyer), but it may become critical in a border-to-border negotiation. However, it may be a solution for dealing with services with a broader scope (for instance, an any-to-any scope), such as a Worldwide Telephony service (based on provisioning with the next hop, as the current PSTN). Another example is a qualitative service for Fast Web Navigation, which gives better treatment for packets sent by some companies or home users.

The resource control function is different for a border-to-border scope, than for an end-to-end scope, as presented in section 3.5. For resource estimation, it takes into account interdomain egress links, instead of destination domains. Similarly, for service offering, interdomain ingress links are also considered, instead of paths between border routers. Consequently, the request and offering matrixes (Figure 3.11 and Figure 3.13) have to be changed to express these differences. The resource provisioning is different and somewhat more complex for a border-to-border scope. The traffic entering a domain from a given ingress link can leave that domain from any egress link. This leads to a hose-like [65] resource provisioning style, which was not considered for the end-to-end negotiation scope as the service scope was deliberately limited to the pipe (1:1) scope (section 1.3), for the sake of simplicity.

Hybrid scope: A hybrid scope is characterized by using different scopes for the WDS negotiation and resource negotiation phases. The WDS negotiation may be achieved using an end-to-end scope and the resource negotiation using a border-to-border scope. This is due to the fact that the WDS negotiation is responsible for guaranteeing stricter definition parameters and it is performed less frequently than the resource negotiation.

5.4.2 Negotiation Control

Negotiation control may be distributed or centralized. It defines who are in charge of the negotiation process and it also refers to the degree of freedom domains have in order to start negotiations.

Centralized negotiation: In this negotiation style there is a central entity, which is in charge of the negotiation process. Whenever a domain wants to engage into a negotiation, it first must contact this central entity. Since it must perform the negotiations on behalf of all CDG participating domains, the central entity may have the ability (depending on the negotiation model) to perform the resource allocation in an optimized way. On the other hand, centralized negotiation may raise concerns on system reliability, once it represents a unique point of failure.

Distributed negotiation: In this style there is no central point of control. Domains are allowed to initiate negotiations at their own discretion, this is not to say that they can start negotiations at any time they want. The negotiation model and/or the CDG may impose some restrictions, in order to organize the entire process.

In a distributed negotiation style, there is no correlation and coordination among negotiations. A buyer domain has no knowledge about other domain’s service needs. A seller domain may receive several unrelated simultaneous service requests, and it has no or little knowledge for taking important decisions involving different domains. For instance, it may fail to determine whether or not the resource allocation is being fair with all buyer domains.

5.4.3 Negotiation Synchronization

Negotiation synchronization defines whether all domains involved in an end-to-end service deployment must be synchronized in order to perform the negotiation.

Synchronous negotiation: Independently of the negotiation scope of control, in a synchronous negotiation, a buyer domain sends a request and expects to receive a response regarding the end-to-end service availability in a fairly predicted time. In this case, “end-to-end” refers to the fact that it is not sufficient to simply receive a response from the next-hop domain.

In a distributed style, all domains within the scope must be ready to process the service request. In a centralized style, the central entity must be ready, having all information at hand required to process the service request.

Asynchronous negotiation: In an asynchronous negotiation, buyer domains do not expect to receive a response when they send an end-to-end service request. They may receive a response from the next-hop or from an intermediate domain, though. The end-to-end response may come later on or even may never come at all. Using this style, domains may not always be sure about the end-to-end service availability. They must rely that the other domains will have a similar predicted behaviour and a good resource provisioning strategy.

5.4.4 Negotiation Timing

Negotiation timing refers to whether successive negotiations must happen in predefined time intervals or whether domains are able to start negotiation whenever they want.

This issue is highly related to resource reservation styles [44]: advance or immediate reservations. On the one hand, advance reservations allocate resources before they will be needed, based on traffic predictions. Thus, the call setup time is significantly reduced and system's scalability is enhanced, by allowing aggregate reservations. Immediate reservations, on the other hand, are made on demand according to instantaneous fluctuations of traffic volumes. Advance and immediate reservations are complementary to each other. The former gives predictability and scalability to QoS-based services, whereas the latter making final adjustments when traffic predictions fail and service activations are blocked by the CAC mechanism.

Periodic negotiation: Periodic negotiations are based on traffic prediction and generate advance reservations. In Chameleon, it is expected that most negotiations will be periodic, i.e., the time interval between two successive negotiations must not be lower than a minimum agreed value established by a CDG. There is, however, no need for the time interval between each pair of successive negotiations to be constant.

Urgent negotiation: If a given service is subject to admission control and an activation request is blocked, then the domain's SB might initiate an urgent negotiation, which can result in an immediate reservation. However, urgent negotiations may not always be allowed by the CDG. Firstly, some negotiation models may not be appropriate for it (for instance, it may be difficult to deal with urgent negotiation in the hierarchical model, as discussed in section 5.7).

Secondly, allowing frequent resource requests and reservations may generate instability in the domain's internal and external resource provisioning. A typical situation wherein an urgent negotiation will be essential is in case of service instability or outage, caused either by a link or router going down, or by an instantaneous network misprovisioning or misconfiguration.

5.4.5 Negotiation Direction

This style is related to the ability of a model to perform negotiations in one or more directions each time a negotiation is triggered. Some services are intrinsically interactive (such as telephony and videoconferencing) and therefore they need performance guarantees in more than one direction to provide the expected behaviour.

Unidirectional negotiation: Services can be negotiated over one direction at a time. This does no imply that, for instance, bidirectional services are not supported. In this case, two different negotiations will be required, for both communication ends.

Bidirectional negotiation: This style permits negotiations for both directions in order for a bidirectional service to happen simultaneously.

Multidirectional negotiation: It is a generalization of the bidirectional negotiation, for supporting broader scopes other than the Pipe (1:1) style. However, it is outside the scope of this thesis (section 1.3).

5.4.6 Negotiation and Routing

Negotiation models may be classified as whether or not they allow the negotiation process to change the interdomain routing.

Routing passive: The negotiation model is routing passive if it does not interfere in the interdomain routing. It only uses interdomain routing information for performing negotiations.

Routing active: When a negotiation model is permitted to interfere in the interdomain routing for domains within a CDG in order to achieve better negotiations, it is referred to as routing active. This style gives higher flexibility to the negotiation, but also introduces some new complexities in the form of changing the way domains update their routing tables.

5.5 Cascade Negotiation Model

In the Cascade negotiation model, each domain is responsible for the connectivity and communication with its immediately adjacent domains[27]. Although there is a global Multi-SLA in the CDG (section 5.3.1) that every domain has to agree upon, the very nature of the specific physical interconnection agreements is let to the discretion of each domain. An end-to-end service is built by concatenating the specific bilateral agreements between each pair of domains.

In this model, a buyer domain willing to sell its users a service that ends in another one, negotiates with its neighbour, which in turn negotiates with its next one, "rippling" messages through until the destination domain and then backwards. Figure 5.3 depicts a negotiation using the Cascade model, with three transit domains between source and destination. In the forward path, messages carry service purchase requests, based on a WDS instance. Negotiation outcomes are communicated to each intermediate SB in the returning path.

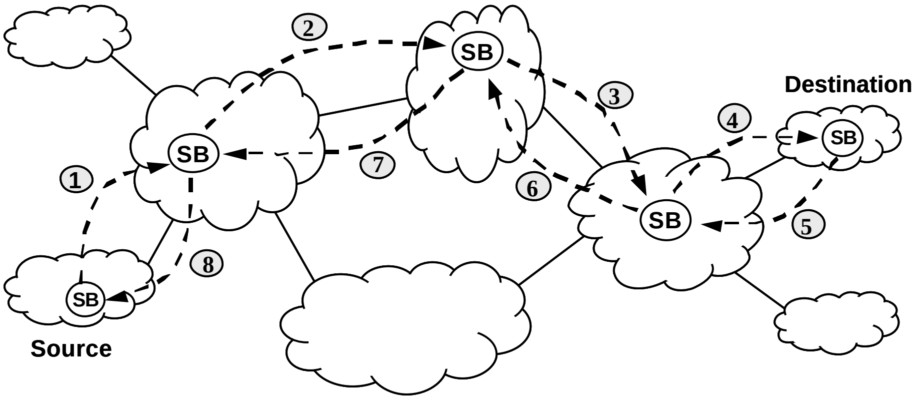


Figure 5.3 – Cascade Negotiation Model

The cascade model raises concerns with respect to scalability, because when the number of domains is high, it generates too many signaling messages [99]. Therefore, it is recommended for use of CDGs of small to medium size (less than 50 domains).

5.5.1 Negotiation Styles

With respect to negotiation styles, the Cascade model may be classified as:

- Scope: Although it relies on bilateral interconnection and decisions, it has an end-to-end scope, because negotiation messages travel from source to destination. After a successful negotiation, a source domain knows whether there will be enough resources made available in the end-to-end path.

- Control: It follows a distributed style, since each domain can autonomously decide when to start a negotiation and its destination domain. A given seller domain may receive several negotiation requests from different sources to a variety of destinations with no prior coordination among them.
- Synchronization: The Cascade model is synchronous since its negotiation is only completed when a source domain receives a response from a previous request.
- Timing: Depending on the CDG, both periodic and urgent negotiations may be supported. The CDG may establish a minimum interval between two successive negotiations (periodic and urgent) in order not to overload domains with too many negotiation requests.
- Direction: It is suitable for unidirectional negotiations, since routes are asymmetric by nature in the Internet. It could only be used for bidirectional negotiations in some very particular situations, probably for small CDGs where domains are confident that they have symmetric routes to each other. Overall, a bidirectional service may be supported by means of starting two different negotiations.
- Routing: The cascade model can operate only in the routing passive style, since it depends on existing interdomain routing tables for exchanging messages.

5.5.2 Implementation Issues

The simplest implementation is by using a protocol such as the Qbone/Internet2 SIBBS [192], where messages flow from source to destination, and return with the negotiation outcome. A protocol for the Cascade model consists of simple request-response messages exchanged by the Service Brokers (SBs) on behalf of each domain. The SBs know each other's addresses by means of the domain discovery function of the service plane. In order to exchange messages in the forward path, every domain has to discover the address of the next-hop domain' SB. These messages must cache all SB's addresses for use by response messages across in the same path, because of the asymmetry of the Internet routing.

[27] The Cascade model was previously called Bilateral model [129][133], because of the bilateral nature of its interconnection agreements and message exchanges.

Although the goal is not to define a protocol for any negotiation model, a general protocol implementing the cascade model could have the following messages:

- WNR (WDS Negotiation Request): In the WDS negotiation phase, this message flows from source to destination verifying the availability of definition parameters (such as delay, jitter and packet loss). In each domain the SB updates the partial values of these parameters (delay and jitter are additive; packet loss is multiplicative [215]), according to its service offering matrix (section Figure 3.13). These parameters must be strictly fulfilled, according to the definition in section 4.3.1. Otherwise, the negotiation fails.

- WNA (WDS Negotiation Answer): This message flows in the backward path, form destination to source, communicating the negotiation outcome: success or failure. If the negotiation is successful, it returns for each parameter its specific negotiated end-to-end value (for instance, a 75 ms delay and a 0.01% packet loss rate). Depending on the service definition, it may also inform the accepted level (section 4.3.1).

- RAR (Resource Allocation Request): Once the WDS negotiation resulted in success, the next step is verifying if there are enough resources (network capacity, also called throughput in the service definition) for that service. Since it is a concave parameter [215], the negotiated throughput is the minimum available in every link of the end-to-end path. The requested resources may be partially granted if it is higher than the minimum end-to-end path capacity. Therefore, in order for a buyer domain not to receive insufficient resources, a real protocol should require from a domain to specify a lower bound for the resource grant. Below that level, the negotiation fails.

- RAA (Resource Allocation Answer): This message flows from destination to source domain, informing each intermediate domain the amount of granted resources. This amount of resources will be used for internal provisioning.

In the cascade model, the service offering matrix is only known internally in a domain. As a domain receives request messages, the SB uses the matrix for checking service availability. Neither the internal service offering nor the administrative polices stating restrictions for service deployment must be known outside a domain.

In order for such a protocol with two phases divided into WDS and resource negotiation to work, a certain routing stability is required. At least, interdomain routing should not change more frequently than the WDS negotiation. Otherwise, these two phases should be combined into only one phase. In this case, the definition parameters should be renegotiated each time the resource negotiation phase is processed. Whether or not a protocol that combines the WDS and resource negotiations is to be used is up to each CDG. The same thinking applies to the other models.

5.6 Hub Negotiation Model

In the Hub negotiation model (Figure 5.4) a domain playing the role of a User Service Provider (USP) is responsible for negotiating with every other domain along the path from source to destination. In other words, the USP represents the buyer domain in the hub model. In a more complex scenario, several USPs may cooperate with each other to build an end-to-end service [100]. As it requires additional agreements and protocols between USPs, only the simplest case with one USP is described here.

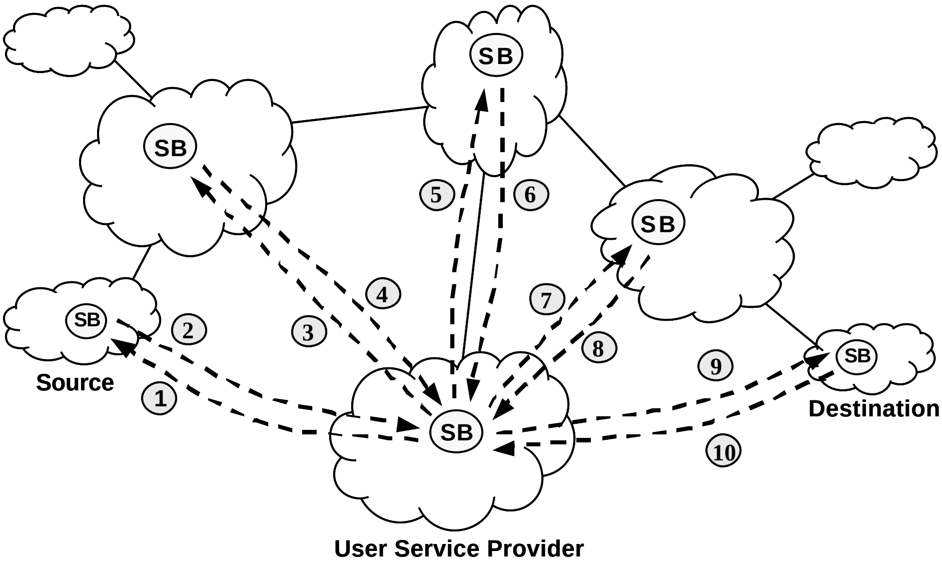


Figure 5.4 – Hub negotiation model

For completing the negotiation, the hub model requires two stages, although Figure 5.4 only represents the messages of one of them for a better visualization.

- Stage 1: It is a pre-negotiation stage, whereby the USP sends messages to all domains involved in the end-to-end service deployment one at a time, checking for service availability. During the resource negotiation phase, domains must pre-allocate resources.

- Stage 2: It is a confirmation stage, whereby the USP sends simultaneously messages to all domains, and waits for their responses. For the WDS negotiation phase, the confirmation merely means that the USP intends to deploy a given WDS through those domains. Deciding whether this stage is necessary for the WDS negotiation phase is up to particular implementation protocols. Regarding the resource negotiation phase, it causes each domain to provision resources for that service.

These two stages are necessary for both the WDS and the resource negotiation phases. Between the two stages, domains must keep the committed resources pre-reserved, that is, they must not grant them to other new service requests.

The Hub model has a clear advantage over the Cascade one. It gives the USP a higher control over the negotiation, because it does not have to rely on other domains to send their negotiation messages. This way, the USP can know exactly the point where the service request is not being fulfilled and may take by itself the decision of whether to decrease the amount of partial granted resources or accept the negotiation failure. However, this comes at the cost of a higher number of exchanged messages, and consequently raises more concerns about its scalability unlike the Cascade model. Actually, it generates twice the number of messages, because of the two stages it defines. Similarly, the time to complete a negotiation is also higher.

5.6.1 Negotiation Styles

With regard to negotiation styles, the Hub model may be classified as:

- Scope: The Hub model has an end-to-end scope, as the USP negotiates with every domain along the path from source to destination.
- Control: It follows a distributed style, even though the USP manages the whole negotiation process. USPs are free to decide when and to which destinations they want to initiate negotiations; they are not controlled by a central entity.
- Synchronization: The Hub model is synchronous since a negotiation is only completed when the USP receives the last confirmation message.
- Timing: Similarly to the Cascade model, both periodic and urgent negotiations may be supported, depending on the CDG policies.
- Direction: It is suited both to unidirectional and bidirectional negotiations. However, the latter require the USP to perform the same process starting from both sides. Hence,

the number of exchanged messages is doubled, and the concerns with the scalability of the model are even greater.

- Routing: Only the routing passive style is permitted, since the USP depends on existing interdomain routing tables for exchanging messages.

5.6.2 Implementation Issues

A protocol for the Hub model consists of simple request-response messages started always by the Service Broker of the USP. The answer messages of the first stage must inform the address of the next-hop domain SB. A general protocol implementing this model could have the following messages for both WDS and resource negotiation:

- WPNR (WDS Pre-Negotiation Request): In stage 1, this message is sent from the USP to every domain verifying the availability of definition parameters for a given WDS in the end-to-end path. Domains do not perform any complex processing upon receiving these messages, they simply consult the service offering matrix and send the information back to the USP. Each domain knows which internal path and interdomain link must be used in order to reach the next-hop domain.

- WPNA (WDS Pre-Negotiation Answer): This is the answer message that follows WPNR. When the USP receives this message, it updates the partial values of the parameters. A domain must also inform the next-hop domain SB address.

- WNC (WDS Negotiation Confirmation): It is a confirmation message (stage 2) informing the WDS negotiation result. The implication is whether or not the USP will involve those domains in the future in resource negotiation for that service in the specific scope verified in stage 1.

- WNA (WDS Negotiation Answer): This message is simply an acknowledgment that domains received the WNC message. Specific protocols may not need to employ it.

- RPAR (Resource Pre-Allocation Request): It is similar to the RAR message of the Cascade model, except that the message flow follows the previous WDS negotiation and the decisions about partial available resources are up to the USP. The domains must pre-allocate the resources they can provide to the end-to-end service and maintain them until they receive the RAC confirmation message.

- RPAA (Resource Pre-Allocation Answer): It conveys the answer of message RPAR with the information about resource availability. Upon receiving the RPAA message, the USP's SB updates the partial available resources control field and sends a RPAR message to the next-hop domain. The destination (last) domain starts the stage 2 upon receiving a RPAA.
- RAC (Resource Allocation Confirmation): After finishing stage 1, the USP informs each domain about the amount of resources they have to allocate for that service.

5.6.3 Star Negotiation Model

The Hub model is more adequate for situations where:

1. The USP is neither the source nor the destination domain, as depicted by case 1 in Figure 5.1.
2. The USP is on the data path between source and destination domains, that is, both endpoints send messages to the USP that in turn forward them to the final destination. As buyer domains must know the traffic volume in order to perform resource estimation, if the USP is not on the data path, the source or the destination domain must send it this information. Although domains are not prohibited to proceed in this manner, it makes the negotiation process more complicated.

The Star model is a slight variation of the Hub model whereby the service buyer is the source domain. This may happen because either the source domain itself plays the role of the USP (Figure 5.1, case 2), or the USP delegated this responsibility to it through a bilateral agreement during the SLA negotiation. Figure 5.5 shows that the Star model requires one less message exchange in each negotiation stage.

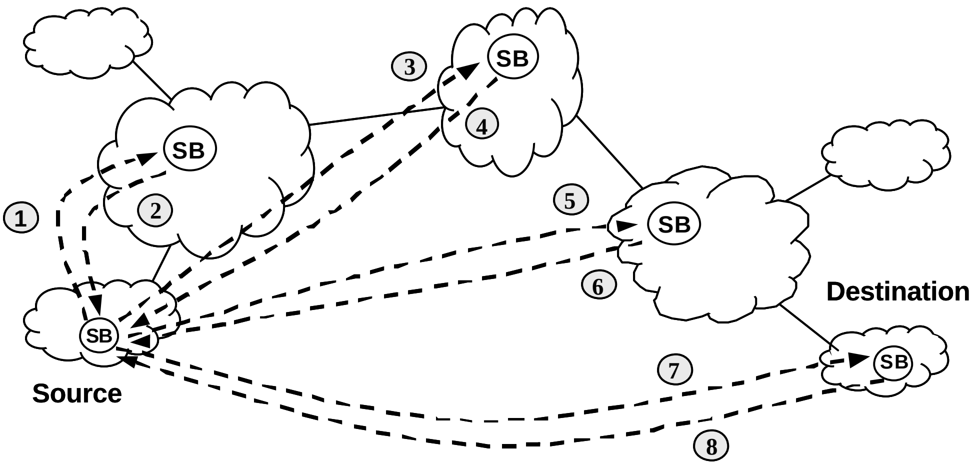


Figure 5.5 – Star Negotiation Model

5.7 Hierarchical Negotiation Model

The hierarchical model (Figure 5.6) introduces the new concept of Service Exchange (SE), which is a central entity responsible for coordinating the service negotiation process among domains of a CDG. The SE performs negotiations on behalf of its participating domains, unlike for instance the cascade and hub models, where negotiations involve only local decisions. When the number of domains in a CDG increases considerably, the SE may be divided up in a hierarchical manner in several other entities that share the work (section 5.7.3).

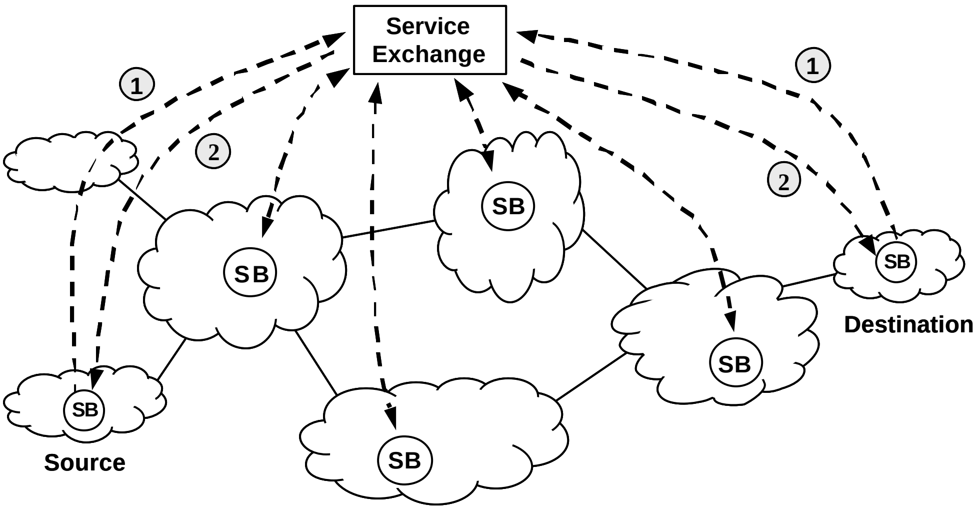


Figure 5.6 – Hierarchical Negotiation Model

During the service negotiation process, SEs seeks at achieving the following goals (some of them may generate conflicts during the negotiation, and the optimization of all of them at the same time may not be always possible):

- Service purchase: most service purchase requests should be accepted.
- Precise reservations: once reservations are done, services should not offer lower QoS to their users caused by sudden lack of the previously negotiated end-to-end guarantees.
- Service sale: most available services should be effectively sold.
- Efficient implementation: implementation has to be efficient, because SEs may experience a high processing load.
- Fairness: negotiation should give equal opportunities for all sellers and buyers.

The hierarchical model is expected to provide some interesting features to the service negotiation, such as efficiency, scalability and fairness (more details in Chapter 6).

5.7.1 Negotiation Styles

Concerning the negotiation styles, the Hierarchical model may be classified as:

- Scope: The scope of the Hierarchical model is end-to-end. Either both source and destination domains are within the area of a single SE, or the SE has to rely on a higher-level SE to complete the negotiation.
- Control: It is only natural that the Hierarchical model should follow a centralized style. The SE decides when to start negotiations on behalf of its member domains.
- Synchronization: It is a synchronous model. Whenever domains receive the results from the SE, it signals that the whole negotiation is completed, regardless of the number of SEs involved in the process.
- Timing: Both periodic and urgent negotiations may be used jointly with the Hierarchical model. However, a CDG most likely will impose limits on urgent negotiations, since the interest of a single domain may affect resources previously granted to other domains. For example, the nature of these limits may not be interfering in any previously negotiated resources, especially when involving domains

not in path of that particular request. In case of interdomain link or router outage, an urgent negotiation can involve all domains in the CDG.

- Direction: There is no restriction for negotiating bidirectional services, even in face of asymmetric routing, since the SE knows all routes in the CDG.
- Routing: Both routing passive and active styles are permitted. The passive mode is the simplest one, since the SE needs not be involved with routing updates. However, the routing active style permits more efficient resource allocation algorithms to be used.

5.7.2 Negotiation Process

In negotiation models that follow a distributed control style, each domain controls its own internal information, trying to obtain the best possible results. In the Hierarchical model, the SE operates in a centralized way as an outsourced entity. Therefore, in order to correspond to domains' expectations, the SE needs to know some information about them. It is mainly related to the CDG "infrastructure", and the SE maintains it up-to-date.

- Available services: The SE needs to know which WDS classes and permanent instances are accepted by the CDG. Although this is not a requirement of the hierarchical model, it is expected that the SE will encompass the function of the CDG Controller (section 3.1.2).
- Interdomain topology: Domains can decide which interdomain links they want to export to the SE. They are not obliged to reveal all their interconnection links, but only those they want to use for deploying advanced services in a given CDG. Regardless the interconnection points being direct-circuit or exchange-based (section 2.1.4), they have to keep the SE informed about CDG topology. Domains communicate to the SE this topology by informing a list of border router (BR) pairs: one of their own and one of a neighbouring domain. BRs represent internal clusters (or PoPs) and their addresses are important information as two domains can have more than one interconnection at distinct points. Based on the address of each domain and their interconnected BRs, the SE builds a topology graph, which is further used for service negotiation.
- Routing: The SE maintains a database containing all routes used within a CDG, in order not to choose routes that are not being used at a given moment. It maintains this

database updated by participating in the interdomain routing using the BGP protocol. The SE uses the routing table in conjunction with the topology graph.

Once the SE has of this information, it coordinates negotiation "rounds" at predefined time intervals, where the negotiations for all domains are done at the same time. Each negotiation round is composed of three distinct steps:

1. Domains send purchase requests and sale offerings to the SE, which are added to the topology graph.
2. The SE performs negotiations using the submitted information. In the WDS negotiation phase, it checks the end-to-end service availability based on the service definition parameters. In the resource negotiation phase, the SE tries to perform the best possible resource allocations, according to the above-mentioned goals.
3. At the end of each negotiation "round", the SE sends the results back to the domains that requested them. The WDS negotiation phase results in a matrix of feasible services between pairs of source and destination domains, in addition to the end-to-end values for the service definition parameters. The resource negotiation phase results in a matrix of resource allocation and signals that resources will be provisioned end-to-end.

Service purchase requests contain the following information: a) service identification: by the WDSID. The negotiated service may be either a WDS class or a WDS permanent instance; b) service definition parameters: in the case of a WDS class, instance parameters must be also informed. On the other hand, in the case of a WDS permanent instance, all the necessary definition parameters are registered in the CDG; c) service negotiation parameters: scope and throughput (resource) are taken from the service purchase matrix. The granularity of the throughput depends on each destination domain. For those domains that declared their internal clusters (PoPs) with the CDG, the throughput must include a list of cluster addresses and individual resource requests. Resource information at the level of clusters is only used to allocate resources within the destination domain. For the other transit domains, it is seen as an aggregate.

Service sale offerings contains the following information: a) service identification by WDSID. No other information about end-to-end services is needed; b) internal paths: domains can use their service offering matrix as a basis for informing the characteristics of their internal path; c) an optional cost associated with each path. Cost is only needed when a CDG allows the SE to use a financial criterion for resource allocation (section 5.7.4).

Domains can choose the granularity level of their internal paths that they want to inform to the SE. A first scenario would be from each BR to every other one. Only domains that have a fine-grain control over their resource provisioning and want to show this information to the SE may use this option. A second possibility is considering the entire domain as having just one path. This high level of aggregation is adequate for small access domains with a simple topology and for domains that rely on over-provisioning as their QoS implementation approach. A third possibility is aggregating paths on a cluster basis and announcing only those main internal paths. A large number of internal paths in each domain contributes seriously for increasing the processing burden on the SE. Therefore, domains should aggregate their internal paths as much as possible.

The upstream domain (the domain that is transmitting packets) is responsible for the sale information for an interdomain link, since it also controls the traffic forwarding on that link. For each link in the topology graph, the SE needs the same information as that of the internal paths.

5.7.3 Hierarchical Relationship of Service Exchanges

The SEs themselves may be organized into several levels in a hierarchical manner. Each SE aggregates the requests where the destination is outside its area and sends them to a higher-level SE, and so on (Figure 5.7). The lower level SE is called SE_1, the next one is the SE_2 and the SE level n is called SE_n. SEs at a same level are not expected to communicate peer-to-peer, because such a situation could lose the envisioned benefits of the hierarchical model (although they are not prevented to do so, notably when there are few SEs at a given level).

The negotiation process in a higher-level SE follows the same idea that of SE_1, but with aggregate information. Similarly to a SE_1, the SE_2 needs information about services and routing which, are common to the whole CDG. However, the SE_2 sees the topology in a higher abstraction level and hence with less details. A node is represented as a domain for the SE_1, whereas from the point of view of the SE_2 a node is a SE_1. Similarly, internal clusters are represented as PoPs for the SE_1, whereas for the SE_2, they are mapped to domains. All SEs_1 belonging to the same negotiation area must inform their addresses and interdomain topology for the SE_2, in the same way domains do it to the SE_1.

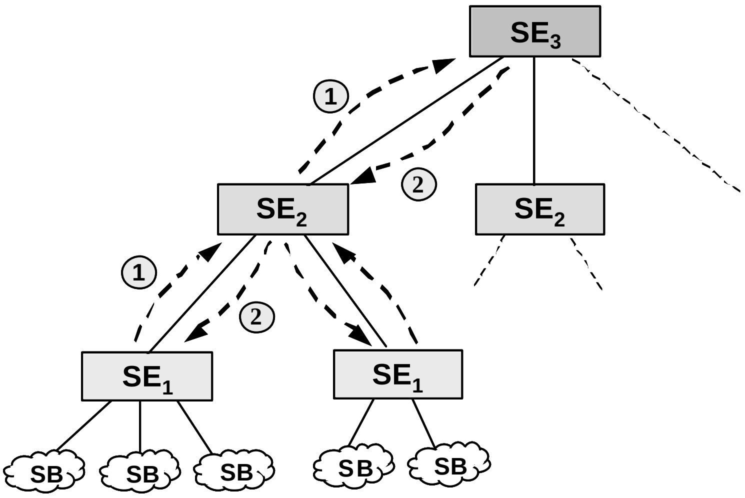


Figure 5.7 – Hierarchical Relationship of Service Exchanges

In a two level hierarchical structure, a negotiation round is more complex than the one for a basic single level structure. Here, eight steps are performed:

1. Within each basic negotiation area, domains send purchase requests and sale offerings to the SE_1.
2. Each SE_1 does its local negotiations. When the destination domain is outside the SE_1 area, a temporary destination domain is chosen among the local domains, in order to make it possible to perform the local part of the negotiation. This temporary destination is the last domain in the path within the SE_1 area, in the border with the neighbouring SE_1 area.
3. Each SE_1 aggregates information about service purchase and sale and send it to the SE_2, taking into account the results of the previous internal negotiation. When dealing with service purchase, if a request is blocked internally, the negotiation is not processed further. When resources are partially granted, the amount of requested resources is adequately decreased for the higher-level negotiation (with the SE_2). For service sale, the amount of resources allocated in the local negotiation is reduced from the offerings for the higher-level negotiation. SEs may also reserve part of the resources for higher-level negotiations, in order to be fair with internal and external requests.
4. The SE_2 performs the negotiation, using the information sent by each SE_1 participating within its area.

5. When the negotiation is over, the SE_2 informs the outcomes to the SEs_1.

6. When a SE_1 receives the responses, it verifies if its requests were blocked or resources were partially granted. In the latter case, the negotiation is reprocessed with the adjusted results.

7. The SE_1 distributes the sold resources to the seller domains and interdomain links. This step is necessary because in step 3 the sale information of the SE_1 is aggregated, including resources available within domains and in interdomain links. Since most commonly only a portion of the available resources are allocated, the SE_1 must divide the aggregated information into a fine-grained information of domains and links.

8. The SE_1 returns all negotiation results to the requesting domains.

With a hierarchy of more than two levels, there will be more steps added between steps 4 and 5. Actually, each new level adds 5 steps to the hierarchical negotiation process.

The issue of purchase and sale aggregation of step 3 is essential to the hierarchical model and deserves more attention. There is a trade-off between aggregation and precision of the QoS guarantees. Aggregation has the well-known benefit of scalability. However, it also leads to loss of fine-grained information and consequently may reflect on the negotiation precision. As far as service purchase is concerned, the main question is the level of granularity a SE should inform to its higher-level SE. If in a given CDG SEs_1 aggregate their requests considering the other SEs_1 as the basic unit of negotiation, then information of the requests for individual domains is lost. For instance, let us suppose that $SE1_1$ and $SE2_1$ are directly connected by an interdomain link between one domain in each SE area. In $SE1_1$ there is a request for a given service of 5 Mbps to domain X and a request of 10 Mbps to domain Y, both in the $SE2_1$ area. If requests are aggregated in a SE_1 basis, then $SE1_1$ has to inform to its higher-level SE_2 a request of 15 Mbps targeted to $SE2_1$. Depending on how the service offering of $SE2_1$ are aggregated, the negotiation results given by the higher-level SE_2 may not be sufficient for providing the expected QoS guarantees for the traffic between $SE1_1$ and $SE2_1$. This would be true in case, for instance, that within the $SE2_1$ area there is a 2 Mbps path to domain X and a 20 Mbps path to domain Y and the service offering is also aggregated.

In order to give more information to the SE_2, SE_1 can present a list of individual requests to their internal clusters, along with the aggregate amount or resources. A cluster may be a domain or an aggregate of domains (this is similar to the SE_1 area, where domains inform a list of clusters represented by PoPs). The SE_1 may inform each domain as a cluster and yet another level of granularity with internal PoPs. This fine-grain approach should provide more precise reservations, but at the same time it would decrease the benefits of having a hierarchy, by imposing a higher processing burden on the SE_2.

To the service sale, aggregation may lead to either underutilization or violation of guarantees caused by network congestion. Similarly to the case of a single domain, a SE_1 may choose to inform its topology in at least three different ways: every internal path, the domain as a single path, or paths aggregated in clusters. Parameters including delay, jitter and packet loss may be aggregated by their maximum, the average or the minimum values. For instance, let us assume that an internal path of a SE_1 crosses a domain that in turn has two possible internal paths, with delays of 10 ms and 30 ms. If the SE_1 chooses the one with 30 ms, its internal path will have a higher delay and therefore some requests may be blocked. On the other hand, if it chooses the 10 ms, then if traffic is forwarded by the 30 ms path, the QoS guarantees may be violated. Aggregating the amount of resources available in each path is also a challenge. If the SE_1 chooses to aggregate using a granularity of SEs_1 with a list of domains (as explained earlier), the information details on PoPs is lost.

The simplest scenario for implementing a multi-level SE organization is by imposing some constraints, such as that each SE covers a distinct geographical area and that a domain is only allowed to take part in one SE area. However, in a more complex scenario, there will be several SEs of the same level in a given region, competing with each other for domains by offering different technical and financial incentives. Domains also may be allowed to participate in two or more SEs. Similarly, there may be several higher-layer SEs competing for lower-layer SEs to participate in their areas. If domains of different SEs happen not to be in the same CDG, another level of agreement is necessary for allowing inter-CDG negotiation (section 5.10).

In the most common case, a CDG will start up with a group of domains willing to use the hierarchical model to deploy advanced services. It is expected that a single SE is able to deal with a maximum of 50 to 100 domains, depending on the interconnection topology, the service request patterns and the complexity of allocation algorithms. In the worst-case scenario, with 100 domains, if each domain sends requests to every other domain, the SE will have to process

10,000 requests at each negotiation round. Based on simulation results, it is not expected that such a negotiation will take more than a few minutes in a well-configured computing system. This is reasonable sufficient when the resource negotiation frequency is daily or hourly. A fact that may cause severe reduction in the number of allowed domains, though, is the use of complex heuristics for resource allocation that have a slow convergence, such as simulated annealing and genetic algorithms.

In any case, when the number of domains in a SE_1 becomes larger than its processing capacity, it must be split up into two or more SEs_1 creating a higher-level SE_2 coordinating the negotiations. The same approach is applicable to the SE_2. Other criteria for splitting up a SE may be geographical location or interest mismatch. The SE tree does not need to be complete, meaning that some branches may be deeper than others. Hence, adopting the simplest scenario of geographical division of SEs, with at most three levels of SEs it is possible to cover the whole Internet. Several countries with a small population of domains can be organized in a single SE_1. On the other hand, countries (such as the USA) with a very high number of domains can be organized with several (tens) SEs_2. One SE_3 is needed for coordinating the SEs_2. Some branches may stretch deeper than three levels, in order to do local arrangements.

5.7.4 Resource Allocation

The hierarchical model is able to perform optimized resource allocations, due to its centralized nature. Since the SE has information about the interdomain topology, service purchase requests and service sale offerings, it can more easily be configured for achieving the negotiation goals than other models [125]. This is especially true when the CDG is operating in a routing active style, letting the SE free for changing interdomain routing tables[28]. Even under the routing passive style, the hierarchical model is able to achieve better allocations than other distributed models, mainly because service requests are processed simultaneously.

When operating in a routing active style, the SE can configure routes in such a way that resource utilization is optimized and resource requests are satisfied as much as possible. Currently, interdomain routing in the Internet is performed by the BGP protocol, based upon a shortest-path algorithm. Although the heuristic used by this algorithm is computationally efficient, it does not optimize network resource utilization, creating a scenario where some

[28] Such optimizations are generally more appropriate when both WDS and resource negotiation phases are performed together

routes are congested while other routes have spare capacity [53]. Other negotiation models, such as the Cascade and the Hub models need to use a shortest-path algorithm, due to the autonomy of each domain in negotiating by itself, although they also allow the utilization of some interdomain QoS Routing techniques [53].

On the other hand, the hierarchical model allows the utilization of algorithms based on efficient heuristics that analyze a whole spectrum of possible solutions, in addition to the more traditional QoS Routing. It was shown that some of these heuristics, such as the Constraint Satisfaction Problem based on backtracking or the Blocking Island paradigm [86], outperform the shortest-path heuristic in intra-domain resource allocation on a basis of 20% to 30% [85]. However, this gain in efficiency comes at the cost of changing the interdomain routing. another source of optimization is by allowing routing decisions to be based on the tuple <*destination, source, service*> and not only on the destination information. This permits a more fine grain allocation of resources, by choosing different routes to different packets heading to the same destination but that come from different sources or services. Once more, this optimization demands changes, well known in the context of QoS Routing. Routing tables must be extended with more fields and entries, increasing substantially their sizes [220]. It also becomes essential the utilization of a modified IGP (Interior Gateway Protocol) capable of distributing route information with different QoS characteristics, such as the QOSPF [14].

Regardless of the heuristics used with the hierarchical model, resource allocation may be optimized according to some criteria, defined by the CDG. The most common examples are:

- Capacity: is defined as the minimum amount of end-to-end available resources (from source to destination), including all interdomain links and internal domain paths. Each time a request is allocated the available capacity of every link and internal path is decreased. Here, capacity is a synonym for throughput and resource.

- Performance guarantees: includes the WDS definition parameters, i.e., delay, jitter and packet loss. Throughput is also a performance guarantee, but it is important enough to deserve a special item (above).

- Financial aspects: is defined as the cost for buyer domains and the revenue for seller domains. Seller domains need to send a cost associated with each internal path to the SE for allowing financial aspects to be considered as a negotiation criterion. The algorithm may choose the lowest cost allocation, the highest revenue allocation, or a combination of both.

In addition to optimization criteria, some constraints, such as routing policies, must also be enforced. Domains may set different types of constraints against other domains, for instance, in order to determine whether or not to be used as transit by some particular domains. The SE can obtain these policies from an Internet Routing Registry [69][145] in case such an approach is recommended by the CDG.

For allocation algorithms based on capacity, an end-to-end route is only selected for a request if no service definition parameter is violated. Within the set of routes that adhere to that condition, there are two groups: routes able to grant totally the resources and those that can do it only partially. When the algorithm tries to finds a global optimal solution, it has a particular criterion for selecting a route (each algorithm has its own criteria). These algorithms are very complex and may execute for a long time before they present the best allocation. In case the algorithm only tries to find local solutions, it may simply rely on a heuristic to select a route. When only routes with partial grants are found, the selected route should be the one that is able to grant the higher amount of resources. When many routes offer the same amount of resources, the decision about which route to select depends on each specific algorithm adopted. Some examples may be[29]: the first route; the last route; the best route, referring to the one with the lower capacity; or the worst route, i.e., the one with higher capacity.

Allocation algorithms based on performance guarantees should for each request, start by finding the end-to-end routes that do not violate the referred guarantees. Among those routes, the one that provides the most favorable conditions is selected, according to some known criteria involving delay, jitter or packet loss. Possible variations for selecting the route are: a) the lowest value for a parameter, such as the delay; b) or the highest value for a given parameter (sometimes this scheme might present encouraging results [131]); c) a combination of those three parameters. Hence, care must be taken in order not to fall into an NP-complete problem, for instance, computing optimal routes subject to constraints of two or three of these parameters is NP-complete [215].

If the algorithm is sufficiently sophisticated, the SE may allow different domains or services to select different optimization criteria. Since the complexity and associated processing time may be a concern in these cases, the SE can do off-line simulations based on previous negotiations for pre-selecting the best routes using different criteria.

[29] These heuristics are well known in the operating systems area [191].

5.7.5 Implementation Issues

The implementation of the hierarchical model requires relatively more effort, mainly for building and configuring the servers responsible for the Service Exchange functionality (including the hierarchical structure). This may involve replication of SE servers for additional reliability. The SE must know the addresses of all SBs (representing domains) and vice-versa. The SBs can be informed of the SE address by manual configuration and then register themselves dynamically. However, in most cases a rather static configuration is likely to be work adequately. Another important aspect is related to the WDS and resource negotiation frequency, which is a decision of the CDG.

The proposal for a general protocol implementing the hierarchical model consists of four types of messages: update, inquiry, request, and answer. “Update” messages flow from domains to the SE and convey configuration or service sale information. Domains are expected to send information regularly to the SE. When the SE does not receive this information, it sends “inquiry” messages to the SBs. “Request” messages are used by domains to inform their service purchase intentions to the SE. The SE communicates negotiation results by means of “answer” messages.

- TIU (Topology Information Update): Domains send interdomain topology information to the SE. TIU messages are needed at the hierarchical negotiation start-up or whenever a change in the topology is detected. Link outages must be reported as topology change when they last for a sufficiently long time for causing violations of the QoS guarantees.

- TII (Topology Information Inquiry): Along with asynchronous TIU messages sent by domains, the SE may explicitly request updates using a TII message.

- SOU (Service Offering Update): Domains inform their service offerings by means of SOU messages. When urgent negotiations are supported by a CDG, domains may inform the SE of sudden service quality variations.

- SOI (Service Offering Inquiry): The SE can also explicitly solicit service offering updates from domains.

- WNR (WDS Negotiation Request): WNR messages are used by domains requesting WDS negotiation to the SE.

- WNA (WDS Negotiation Answer): The SE communicates WDS negotiation results using WNA messages.
- WNI (WDS Negotiation Inquiry): The SE can also explicitly solicit WDS negotiation requests from domains.
- RAR (Resource Allocation Request): With RAR messages, domains send their resource needs in order for the SE to process the resource allocation.
- RAA (Resource Allocation Answer): RAA is the message sent back by the SE for communicating the resource negotiation results.
- RAI (Resource Allocation Inquiry): The SE can also explicitly solicit resource negotiation requests from domains.

The same messages can be used for negotiation among SEs at a higher-level relationship, as described in section 5.7.3. Routing messages are not included here as they are exchanged by the BGP protocol.

5.8 Wave Negotiation Model

The Wave negotiation model is driven by service offerings, unlike the previous ones, which are demand-driven (i.e., driven by service requests). Seller domains send service offerings to their adjacent domains, which are valid for a certain period of time. The same resources are not offered to other domains while a service offering has not expired. Upon receiving an offering, a domain can either accept or reject it by simply ignoring it. In case of a seller domain, the service offering scope can be extended by including its local service offering, if the definition parameters are not violated in the end-to-end scope. The amount of resources in the service offering may be decreased according to each domain's previous resource estimates and service acceptances. In turn, this domain sends an offering to its adjacent domains. A domain may receive several offerings (from different domains) covering the same scope, i.e., to the same destination. In such a scenario, the seller must decide which offering it will use for extending the end-to-end service based on internal policies and heuristics [74].

When a buyer domain receives an offering, it can be accepted if some conditions are fulfilled: a) the domain is interested in buying that service (this information is in the resource request matrix); b) the definition parameters are not violated in the end-to-end path (considering its network as well); and c) it is the best offering involving the desired service and destination (when it receives two or more offerings). When a seller domain receives the "accept" message, it may confirm or reject it, in case the service offering expired or an unexpected error was detected.

Figure 5.8 depicts an example of the negotiation process of the Wave model. Domain D8 sends an offering to domains D4 and D6. D6 accepts it and extends it with its local service offering, deciding to divide the resources up into two different offerings and sending them to domains D4 and D5. D4 receives two offerings with destination D8 (from D8 itself and from D6). It then chooses the direct path to D8 and extends it to D3. D5 accepts the offering from D6 and also extends it to D3. In turn, D3 chooses the offering from D4 and extends it to D1, which finally accepts the offering, hence ending the negotiation. The negotiation result is that the end-to-end path from D1 to D8 is D1-D3-D4-D8.

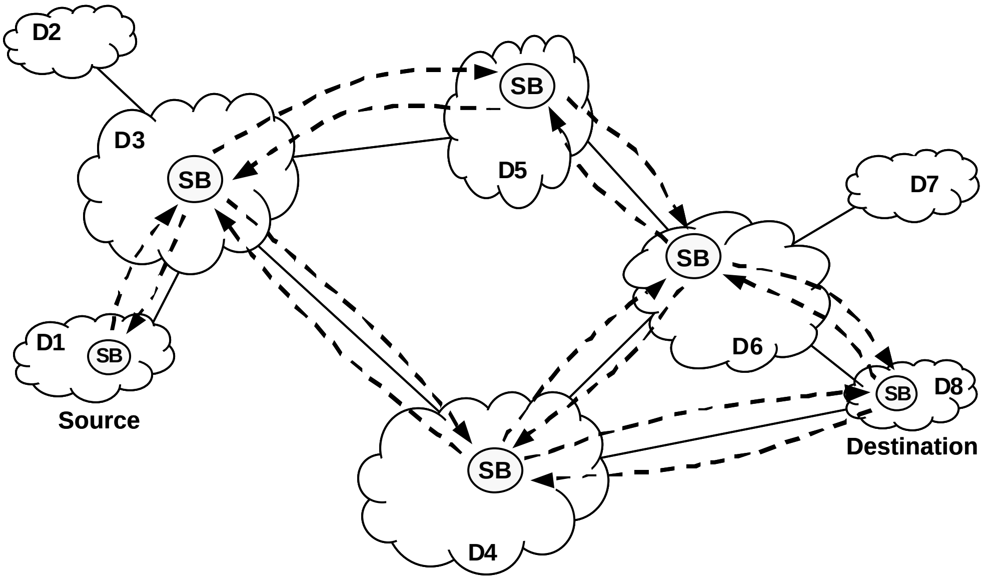


Figure 5.8 – Wave Negotiation Model

In the Wave model, resources may be wasted if a seller domain accepts a service offering and has no success in selling it further. Should this happen, the domain has to retain those resources until the next negotiation even though they will not be used at all.

5.8.1 Negotiation Styles

With respect to the negotiation styles, the Wave model may be classified as:

- Scope: The scope of the Wave model is end-to-end, at least from the service buyer standpoint. Receiving a service offering is a firm guarantee that the service will be available along the end-to-end path. On the other hand, when a seller domain receives an acceptance from another intermediate seller domain, it does not know if the end-to-end service negotiation has successful completed.

- Control: Control is distributed in the Wave model, since each domain can take its own individual decision on service offering and acceptance.

- Synchronization: It is an asynchronous model, as only the last intermediate service seller and the service buyer must be synchronized in order for the negotiation to be completed.

- Timing: Similarly to the Cascade and the Hub models, both periodic and urgent negotiations may be permitted.

- Direction: The Wave model is only suited to unidirectional services, since only service offerings are sent from service sellers. By the way, the traffic direction is opposite to the direction of offering messages. In order to allow bidirectional services, the model must be changed to combine both service offering and request in the same message. Even so, it would only work under symmetrical routing.

- Routing: It uses the routing active style, since a domain may choose one from many service offerings coming from different domains. In that sense, it implements a type of QoS Routing and must interact with the BGP protocol in order to exchange routing information. If the routing passive style is retained, a small change in the model is required. A domain should only accept a message if the seller domains is the downstream domain to the service destination.

5.8.2 Implementation Issues

The Wave model assumes that the WDS and resource negotiation phases are combined. A general protocol may have the following messages:

- SO (Service Offering): Service sellers send SO messages to their adjacent domains offering services with the scope limited to domains between them and the destinations.

- SA (Service Accept): Service buyers or intermediate seller accept service offerings and send SA messages.

- SC (Service Confirmation): Upon receiving SA messages, service sellers send SC messages to confirm the negotiation. Should an unexpected problem happen, sellers send SRJ (Service Reject) messages, instead.
- SR (Service Request): This is an optional message, used by service buyers to solicit that an urgent SO message be sent.

5.9 Border Negotiation Model

The Border model is based on bilateral message exchanges between two adjacent domains, mainly for resource allocation. All domains play the role of both buyers and sellers. Figure 5.9 illustrates this scenario, with request messages being sent (1) and the response messages being returned (2). For purchasing services, domains send service request messages to their neighbours, which also represent the set of all possible destinations. When service is granted, buyer domains cannot rely on the fact that the QoS guarantees extend beyond their next-hop domains. As a result, the border model is more suitable for the resource negotiation phase.

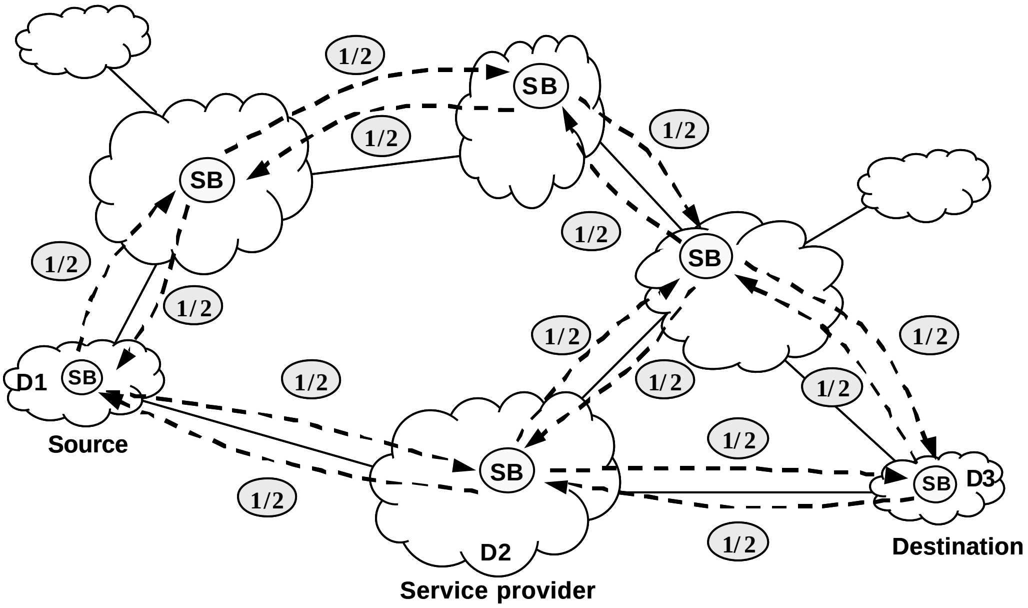


Figure 5.9 – Border Negotiation Model

During the WDS negotiation phase, i.e. when negotiating service definition parameters (delay, jitter, packet loss), there are two possibilities. The first one is limiting the service scope to at most three consecutive domains, which means a buyer and two neighbours. The second one is relying on another model for the end-to-end WDS negotiation and using the Border model only for resource allocation. Although the guarantees are border-to-border, the correct negotiation (and provisioning) between each pair of domains along the path from source to destination ought to produce end-to-end guarantees in a stable environment [199].

5.9.1 Negotiation Styles

With respect to the negotiation styles, the Border model may be classified as:

- Scope: As the name implies, in the Border model the scope is border-to-border. As such, resource control functions are simpler (section 5.4.1), since the far-reaching resource estimation function of the end-to-end scope is no longer necessary. If the WDS negotiation phase employs a different model with end-to-end scope (this possibility was raised above), a hybrid scope is built.
- Control: Each domain is able to take negotiation decisions individually. Therefore, a distributed control style is used.
- Synchronization: The Border model follows an asynchronous style, since only neighbouring domains have to be synchronized to process the negotiation.
- Timing: It supports both periodic and urgent negotiations, depending on the specific bilateral agreements between each pair of domains (within the SLA negotiation phase).
- Direction: The Border model can be used for negotiating both unidirectional and bidirectional services. However, allowing bidirectional services to be negotiated straightforward may not be worth the trouble, since it introduces additional complexity and fails to produce end-to-end guarantees anyway.
- Routing: Only the routing passive style is supported. SBs do not have enough information for interfering with interdomain routing.

5.9.2 Implementation Issues

The implementation of the Border model is much simpler than the other previous ones. A protocol should have only two messages (request and answer) for the WDS negotiation phase and three messages for the resource negotiation phase (request, answer and confirmation).

Answering a request is also not so complex, because domains only rely on their own internal resource control function. A general protocol for implementing the Border model could have the following messages:

- WNR (WDS Negotiation Request): If the scope of a service is limited to two or three consecutive domains, a buyer domain can use WNR messages to negotiate service definition parameters with its neighbours. For example, in Figure 5.9 assuming that domain D2 wishes to provide a service from domain D1 to domain D3. D2 sends WNR messages for both, waits for answer messages (WNA), checks if the service guarantees are maintained end-to-end and (if this is the case) sends confirmation messages (WNC) for D1 and D3.
- WNA (WDS Negotiation Answer): Seller domains annunciate their service availability to buyer domains by sending WNA messages.
- WNC (WDS Negotiation Confirmation): Buyer domains confirm the negotiation completion by means of WNC messages.
- RAR (Resource Allocation Request): Buyer domains send resource requests to their neighbouring seller domains using RAR messages.
- RAA (Resource Allocation Answer): RAA is the answer of the seller domain. In case resources are (totally or partially) granted there is no further confirmation message from the service buyer.

5.10 Inter-CDG Negotiation

The goal of the Chameleon architecture is deploying advanced services in the entire Internet. Although some negotiation models, especially the hierarchical one, are supposed to provide the required technical and financial incentives, it is not likely that all domains in the Internet will be associated to only one CDG. Therefore, mechanisms for enabling possibly several different CDGs to interoperate are needed. Another situation requiring inter-CDG negotiation is that of domains refusing to establish multilateral SLAs and opting for creating bilateral SLAs with every other domain. This scenario may be represented in Chameleon by the number of CDG members being as small as two domains. In any case, the main challenges for inter-CDG negotiation are at the level of business and political matters. Technically, the main constraint may be the variety of different Local Services (LS) deployed by CDGs. This may be minimized if CDGs choose to implement only Well-Known Services (WKS).

In order for CDGs to interoperate, there must be at least one interdomain link between them. This can be obtained either by one domain being member of two CDGs or by having a link between two domains belonging to a different CDG. In a given CDG a domain that is responsible for the inter-CDG connection must be able to provide transparency to the negotiation process for both sides, making adaptations whenever these are necessary. Therefore, this domain behaves internally similarly to a proxy, representing either the source or the destination domain of the other CDG. Depending on the configuration adopted, the operation of the negotiation model must be changed as well.

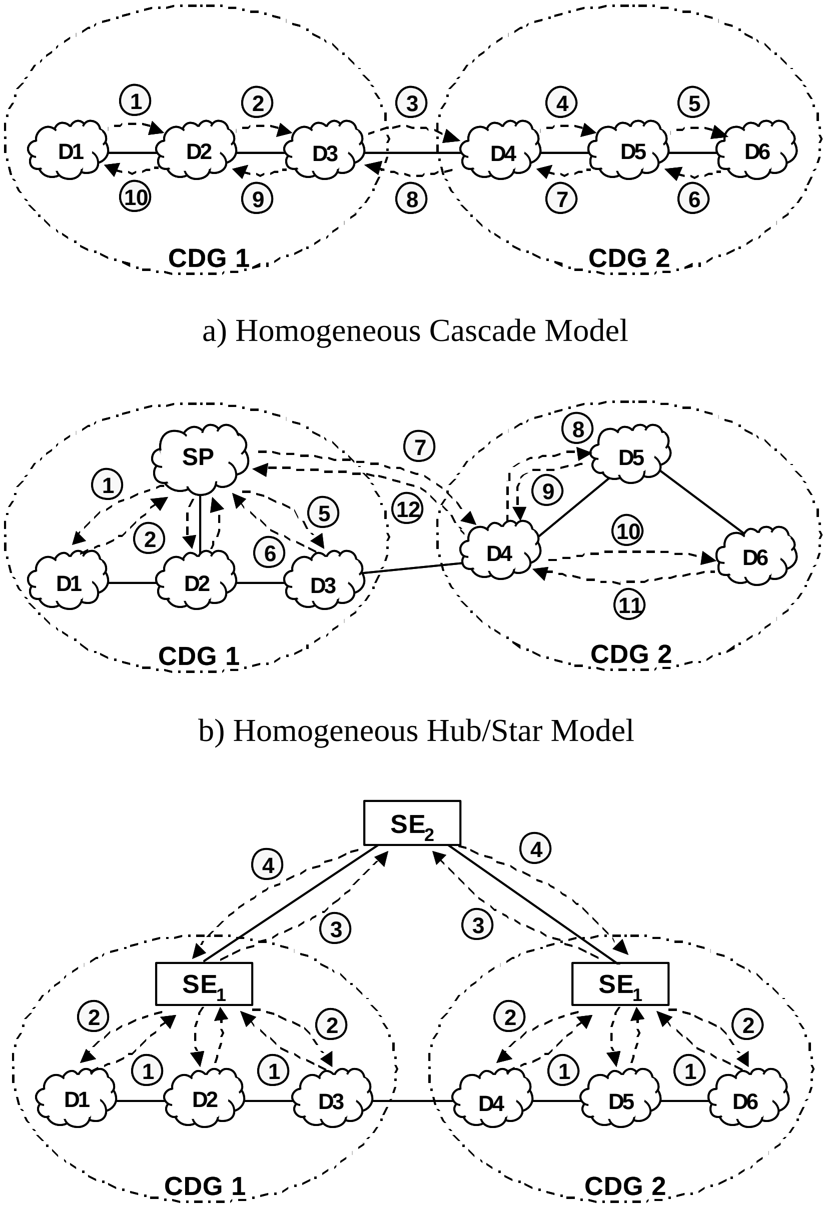
When CDGs seek to collaborate with each other for deploying services, they might face two different situations when it comes to negotiation models: homogeneous or heterogeneous inter-CDG negotiation. In other words, they may employ the same model or different models. Homogeneous inter-CDG negotiation is easier to deploy, but it may not be always possible, as CDGs can be using different models for a long time. On the other hand, for heterogeneous inter-CDG negotiation, certain combinations of models may be difficult to implement (and obtain guarantees from). That is the case of combining negotiation models scope with end-to-end and border-to-border scopes, which yields an intermediate scope that may become difficult to manage.

5.10.1 Homogeneous Inter-CDG Negotiation

Figure 5.10 depicts the interoperation of two CDGs for all of the five negotiation models described in this chapter. In those scenarios, domain D1 is the source that initiates the negotiation and domain D6 is the destination. The bullets indicate the sequence of messages needed for completing a negotiation. For the Cascade model, the inter-CDG negotiation is fairly simple (Figure 5.10a). Domains D3 and D4 simply forward messages to each other in the same way they proceed in an intra-CDG negotiation. When there are more than two CDGs, the same process is done recursively.

For the Hub model, some adaptations are needed (Figure 5.10b). Although any configuration should work correctly, if only the source CDG (CDG1, in this case) uses the Hub model and the other use the Star variation, the message flow follows a shorter path. The USP of CDG1 negotiates normally with domains D1, D2 and D3. However, for completing the end-to-end path, it relies on domain D4, which is playing the role of the USP for CDG2. The SP sends a message to D4 and waits until it completes the negotiation.

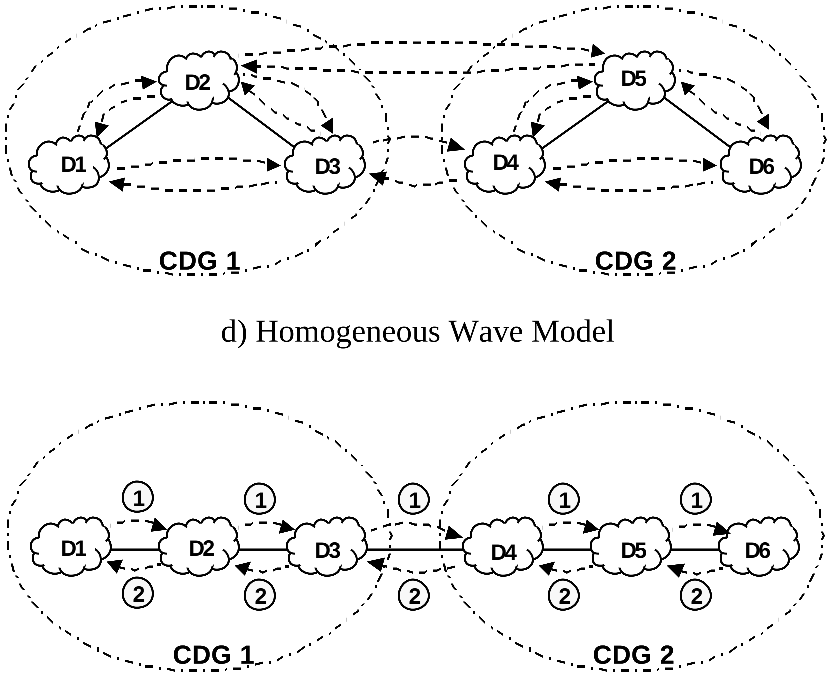
The inter-CDG negotiation for the Hierarchical, Wave and Border models is also simple. For the Hierarchical model (Figure 5.10c), CDG1 and CDG2 only have to agree on a common SE_2, which may be either located at one of the involved CDGs or an outsourced third-party. For the Wave model (Figure 5.10d), no special adaptation is required, although this model will present better results in case there is more than one interconnection link between CDGs. Finally, for the Border model (Figure 5.10e) the inter-CDG negotiation is quite natural, since by its own design the negotiation scope extends only up to neighbours of a domain.



a) Homogeneous Cascade Model

b) Homogeneous Hub/Star Model

c) Homogeneous Hierarchical Model



d) Homogeneous Wave Model

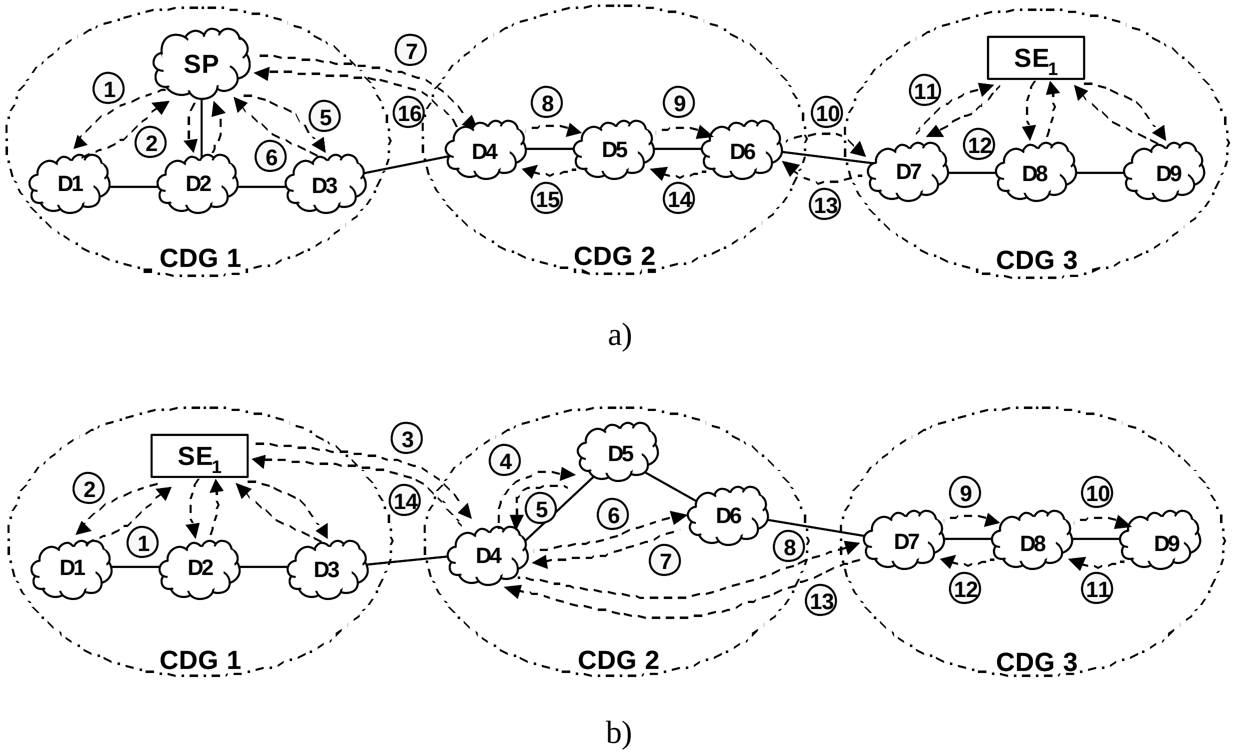
e) Homogeneous Border Model

Figure 5.10 – Homogeneous Inter-CDG Negotiation

5.10.2 Heterogeneous Inter-CDG Negotiation

In a heterogeneous inter-CDG negotiation, care must be taken in order to be sure that the combination of domains will produce the expected guarantees for all involved CDGs. It is not likely that every random combination of models will always work. For instance, combining the Wave model, which is driven by service offerings, with the other demand-driven models may lead to some unexpected results. Similarly, combining the Border model with models that have end-to-end scopes will yield intermediate scopes. CDGs are not expected to accept not having end-to-end guarantees. Therefore, in this section only the Cascade, Hub and Hierarchical models are considered.

Two possible configurations are envisioned in a heterogeneous inter-CDG negotiation: flat and hierarchical. Figure 5.11 depicts two scenarios for flat configuration. In the first one (Figure 5.11a), CDG1, CDG2 and CDG3 use the Hub, Cascade and Hierarchical models, respectively. The Cascade and Hub models behave the same way when used in a homogeneous configuration. The Hierarchical model behaves as in a single CDG negotiation, with domain D7 acting as a proxy. In the second scenario (Figure 5.11b), the order of models is Hierarchical, Star and Cascade. It shows that the Hierarchical model needs some changes when its CDG is not the destination one. When the SE detects that the destination is out of its area, instead of forwarding the message upwards to the higher-level SE, it sends them to the first domain in the path of the next-hop CDG. Hence, CDGs are required to allow urgent negotiations. Otherwise, a negotiation request could last too much time to be completed, and consequently times out.



a)

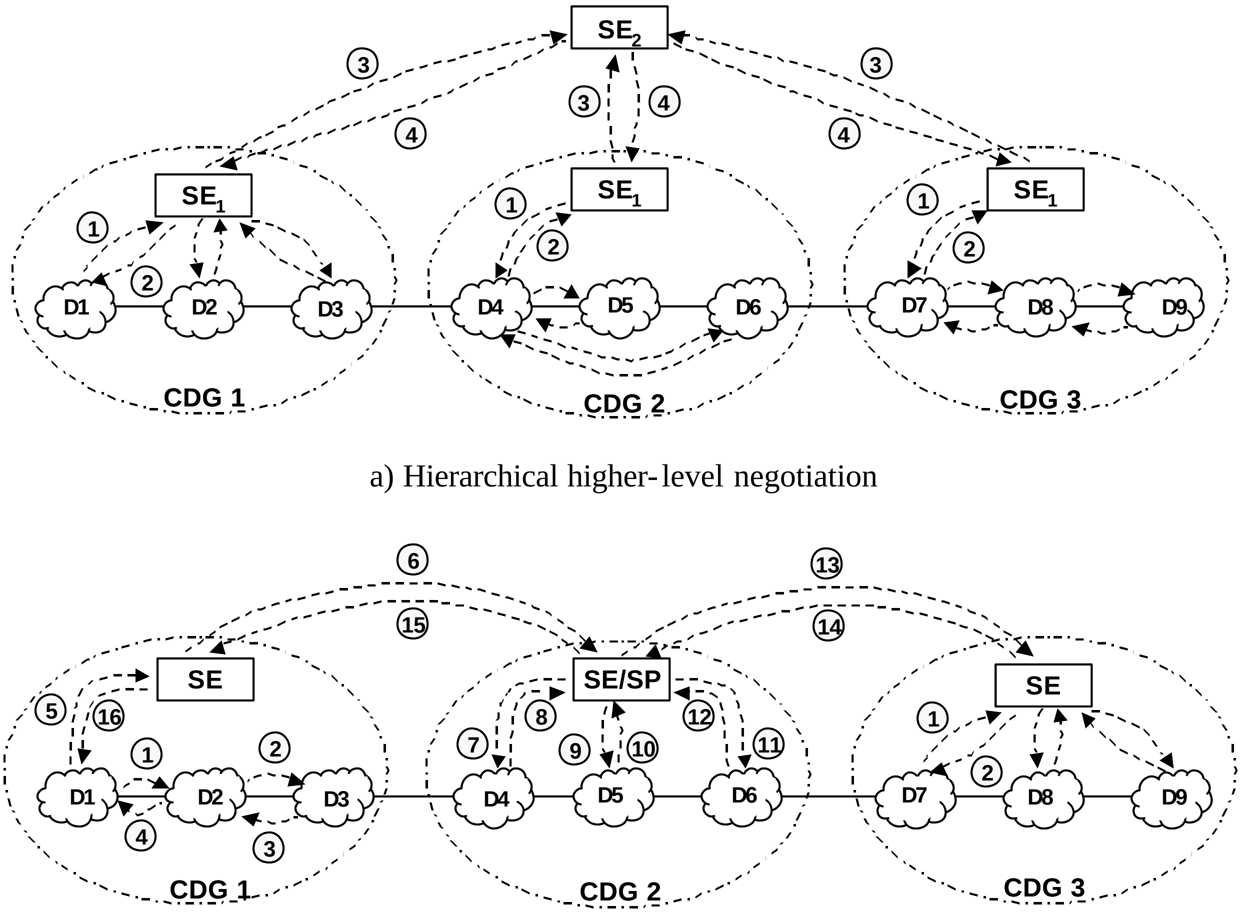
b)

Figure 5.11 – Heterogeneous Inter-CDG Negotiation – Flat Configuration

CDGs using different negotiation models may also interoperate by means of a hierarchical structure. To put it more simple, every CDG needs an entity similar to the Service Exchange, even if it does not perform intra-CDG negotiations, but only for aggregating service requests and offerings for inter-CDG negotiations. The hierarchical configuration may or may not rely on the hierarchical model to perform higher-level negotiations.

Figure 5.12 depicts two scenarios for a hierarchical configuration. In the first one (Figure 5.12a), the hierarchical model is used. CDG1, CDG2 and CDG3 use the Hierarchical, Star and Cascade models, respectively. For the two latter, domains have to send service request and offerings to the SE. When they are initiating an inter-CDG negotiation, domains proceed normally for the internal part of the path. For the remaining path, the source domain relies on the SE. Once the negotiation reaches the SE_2, it must be able to complete the negotiations as in a pure hierarchical model, in order not to loose its scalability benefit. Therefore, CDGs implementing the Star or the Cascade models need to behave as in the hierarchical model, for inter-CDG service offering.

In the second scenario (Figure 5.12b), the Cascade model is used for higher-level inter-CDG negotiations. Domain D1 triggers a negotiation using the Cascade model in the CDG1 area. It transfers the responsibility to the SE for the inter-CDG negotiation. The SE negotiates with the other SE of CDG2 using the Cascade model, which also plays the role of the User Service Provider and in turn negotiates with the member of CDG2 using the Hub model. The request is further passed to the SE of CDG3 which uses the Hierarchical model and sends the negotiation results back to CDG1.



a) Hierarchical higher-level negotiation

b) Cascade higher-level negotiation

Figure 5.12 – Heterogeneous Inter-CDG Negotiation – Hierarchical Configuration

5.11 Related Work

Interdomain service negotiation is a quite new research area, but it has yielded considerable discussion in the last few years. However, to the author's knowledge, a work similar to the one presented herein is yet to be found in the literature. The approach adopted in this chapter is of separating user from transport service negotiation, characterizing the roles in an interdomain negotiation, proposing a classification of negotiation models according to some styles, presenting together five different negotiation models and proposing solutions for the interoperation of different negotiation models.

Most of the existing work is focused on resource reservation for DiffServ networks. In Chameleon, not only network resources (i.e. throughput) can be negotiated. It is aimed at negotiating services based on other QoS performance parameters, including delay, jitter and packet loss. Chameleon also permits the utilization of any QoS provisioning mechanisms, not only DiffServ. The five negotiation models presented in this chapter are relatively simple adaptations of existing approaches to the Chameleon Architecture, except for the Hierarchical model, which deserved more attention and was described in more details.

The Bandwidth Broker (BB) proposed by Van Jacobson et al. for the Two-Bit Differentiated Services Architecture [151] was the pioneer concept for resource negotiation and provisioning, although strongly directed to DiffServ networks. In the Internet2 QBone project, a BB architecture was developed and a protocol for interdomain resource negotiation was specified [192]. The Cascade negotiation model may be seen as an extension of the BB model.

Negotiation models also appear as organizational models for operator interconnection in EURESCOM projects. Particularly for IP QoS services, EURESCOM P1008 [70] is a finished project that proposed the use of the Cascade, Hub and Star models [100]. However, this work does not elaborate further details on service negotiation issues and protocols.

In the TEQUILA project, two models for interdomain SLS negotiation based on the Cascade model have been proposed [94]. The Hop-By-Hop SLS negotiation is similar to the Cascade model whereas in the End-to-End SLS negotiation, it is supposed that a pipe from source to destination have been negotiated previously, by whatever means. In TEQUILA, the SLS negotiation request is originated by the service customer. Hence, when an SLS negotiation request reaches a domain that has a pre-established pipe with spare capacity to the same destination, no further messages are necessary. TEQUILA also has defined and implemented a

Service Negotiation Protocol (SrNP) [211], mainly intended for customer-to-domain negotiations. On the other hand, in Chameleon only domains associated to a CDG are able to negotiate transport services. If a customer (end-user) wants to participate, it needs to implement the Chameleon architecture similarly to the other domains and be a member of a CDG. However, normally customers do not need to do it, as they can rely on their transit providers.

The AQUILA project has developed a protocol called BGRP Plus [174] for interdomain negotiations. It is an extension of the Border Gateway Reservation Protocol (BGRP) [159], which uses a sink-tree approach based on aggregations for getting scalability in interdomain resource reservations. The BGRP Plus assumes that a service is always described by means of a Global Well-Known Service (GWKS). This is similar to the Chameleon approach, but is different from the majority of the other proposals, which either do not give much attention to service definition or assume that a generic SLS is being negotiated.

The Clearing House (CH) architecture [44] for QoS provisioning was the first motivation for the Hierarchical model. The CH is based on the key concepts of hierarchy and aggregation. The Service Exchange and Service Broker are called Clearing House and Logical Clearing House (LCH) respectively. However, it is limited to resource negotiation, that is, it is not able to identify services and perform negotiations based on different QoS parameters.

Hierarchy is a very common concept, used in a variety of areas for solving different problems. Therefore, it is important to establish the fact that the Hierarchical model, in the same way as the other models, is aimed at performing transport interdomain dynamic service negotiation. As such, it is completely different from other approaches that use hierarchical BB architectures for resource allocation in the "intradomain" sphere. Some examples are the Resource Management Agent (RMA) of the AQUILA architecture [163], the Multiple Bandwidth Broker (MBB) architecture [221] and the hierarchical BB proposal for implementing the Resource Mediator in CADENUS [51].

The Wave negotiation model is an adaptation of the SLA Trader (SLAT) [74], which is an enhancement of the BB for service negotiation and pricing for DiffServ networks. It has also been defined a protocol for the SLAT, called SLATP. Another proposal focused on pricing is the Resource Negotiation and Pricing Protocol (RNAP) [214], which gives more attention to the negotiation of multimedia services between the user and the network (although it can be used among different networks as well).

The Two-Tier architecture [199] proposes a resource management model for the Internet based on the BB model that clearly separates intradomain and interdomain resource allocation activities. For the former, it uses a modified version of the RSVP protocol. The latter is based on a border-to-border style, where resources are negotiated in an interdomain link between two neighbouring domains. The decision for increasing the amount of reserved resources in an interdomain link is taken whenever the traffic exceeds a high watermark value. However, for decreasing the resources, the BB of the upstream domain uses a hysteresis process, to be sure that the traffic volume also decreased in a persistent way. The Border model is inspired on the Two-Tier architecture.

The COPS-SLS [149] is an extension of the COPS protocol for intra- and interdomain service level negotiation for domains that implement policy-based networking. The PDP (Policy Decision Point) and PEP (Policy Enforcement Point) components of COPS are adapted to SLS negotiation and called SLS-PDP and SLS-PEP. For the intradomain negotiation, the SLS-PDP represents the network provider and the SLS-PEP the customer, whereas for the interdomain cases the two involved domains have SLS-PDP and SLS-PEP components. The approach adopted by the COPS-SLS protocol makes it operate in a border-to-border style.

5.12 Summary

The work on dynamic interdomain service negotiation for the Internet is very recent. Most of the existing proposals focus on the negotiation between a provider and its client corporate networks, since it is a short-term necessity of the ISPs for automating their service offerings. The work achieved in Chameleon can positively contribute to the development of solutions for the negotiation among domains of the Internet backbone. The most distinctive features of the Chameleon's approach for service negotiation are:

- The negotiation process is comprised of the SLA, WDS and resource negotiation phases. It is useful for putting in the right place the main issues related to this subject.
- The classification of negotiation models based on negotiation styles. This simple taxonomy is useful for comparing the features, limitations and the applicability of the negotiation models.
- The proposal of the hierarchical negotiation model, which is seen as the most promising one (also corroborated by the simulations of the next chapter). The

hierarchical model was therefore described with a higher level of detail than the other models.

- Adaptation of the other four models to the Chameleon architecture. The cascade, hub, wave and border models were developed having in mind slightly different scenarios, but can be fitted to work with the Chameleon model.
- Inter-CDG negotiations involving different models, when two or more CDGs need to interact with each other for deploying advanced services in a broader topological scope.

In a simple and rough comparison of the five negotiation models, the following impressions can be initially pointed out:

- The cascade model is very simple and effective. It checks service availability along the whole path between source and destination domains, thus yielding an end-to-end scope. However, it is expected to face scalability problems, mainly in large CDGs, due to the high number of signaling messages that it generates.
- The hub model gives higher control to the Service Provider over the negotiation. On the other hand, it also raises scalability concerns, because it requires twice the number of messages than the cascade model. Both models have been used by network operators, though in a static way, for concatenating services in the current Internet.
- The hierarchical model, proposed in this thesis, is expected to outperform the other two models in criteria such as efficiency and scalability. A distinctive feature is that messages are not exchanged by entities at the same level.
- The approach of the wave model is driven by service offerings, unlike the other ones. It is based on exchanging messages among SBs of neighbouring domains, but they are not targeted to particular destination domains, as for the cascade and hub models. Therefore, a great deal of unnecessary messages may be generated, thus also limiting its scalability.
- Finally, the border model is the simplest one, since domains exchange messages only on a bilateral basis and it does not require any centralized entity for performing the negotiations. However, this approach yields a border-to-border scope where no firm end-to-end performance guarantees can be provided.

An in-depth evaluation of the cascade, hub and hierarchical models is undertaken in the next section, thus providing more conclusive results.

Chapter 6

Evaluation of Negotiation Models

The goal of this chapter is comparing negotiation models according to some selected criteria. The five negotiation models presented in Chapter 5 have distinctive features, which make it difficult to adopt conclusions about their suitability to different scenarios, unless a sound performance analysis study is undertaken. In this chapter, only the cascade, hub[30] and hierarchical models are compared, because of their end-to-end characteristics, following the same reasoning of section 5.10.2. The criteria of efficiency and fairness were most extensively used for comparing these negotiation models, through a simulation-based evaluation.

Section 6.1 describes the configurations used in the simulation study. Section 6.2 evaluates the Gaussian predictor used in the simulation for generating the resource estimation matrix (section 3.5.1). The simulation results, along with a short analysis, of the criteria of efficiency and fairness are found in sections 6.3 and 6.4. An analysis of the scalability criterion, corroborated by simulation results is presented in section 6.5. Section 6.6 presents simulation results for a configuration of the hierarchical model with 2 levels of Service Exchanges. Section 6.7 compares the criteria of reliability and resilience, financial incentives, and complexity and costs. Section 6.8 summarizes this chapter with the main conclusions taken when the results of the different criteria are analyzed as a whole.

[30] In reality, the star variation (section 5.6.3) was evaluated, although further references will always mention the "hub" model.

6.1 Simulation Configurations

As the main simulation platform, the Network Simulator (*ns*) [148] was used, and extended to implement the functionalities that are necessary to carry out service negotiation in the Chameleon architecture.

The standard random number generator distributed with *ns*, the Park-Miller [146], was changed to the Mersenne-Twister [142] generator. Firstly, Mersenne-Twister provides a maximum period of $2^{19937} - 1$, so that generation of repeated number sequences is not likely to happen (Park-Miller's period is $2^{31} - 2$). Secondly, its implementation is efficient because it is based on fast arithmetic operations (no multiplication and division). Finally, Mersenne-Twister passed spectral tests for randomness. These features justify changing the *ns* native random number generator from Park-Miller to Mersenne-Twister in an attempt to make simulations run faster and produce more reliable results.

6.1.1 Services and Traffic Models

The simulation study refers to a very simple WDS supporting interactive voice and video applications. The service definition is presented below, which is a mix of a WDS class and a WDS instance.

- Service identification:
    - Service name: Sample Multimedia Service (SMS)
    - WDSID: 000100
- Scope: the pipe style is supported as (*source domain: destination domain*).
- Performance guarantees: two parameters are defined for SMS.
    - Delay: one-way delay, with a higher bound of 200 ms.
    - Throughput: this is the resource to be negotiated; the lower bound for resource grant is 0%.
- Direction: SMS is a bidirectional service.

A real service would need a more complete definition. However, as the purpose of this service is only allowing the evaluation of service negotiation models, assigning more constraints to it would not change the validity of the obtained results. In order to participate in service negotiations, domains must inform the WDSID, the destination domain and the amount of requested resources.

Two user services were employed to generate traffic for this evaluation, namely voice and video services, which were simulated separately. For the voice service, the call arrival rate in each domain D_i is modeled as a Poisson process of intensity l_i calls per second, where l_i is the mean. The call duration is exponentially distributed[31] with a mean of $1/m = 120$ seconds, where m is the rate parameter. The traffic load arriving at each domain is defined as $r_i = l_i/m$. Voice sources are modeled as an On-Off Markov process, which alternates between activity and inactivity periods. The duration of these "on" and "off" periods is exponentially distributed with a mean $a = 1.004$ and 1.587 seconds, respectively. Each source generates CBR traffic at 80 Kbps when "on" and 0 Kbps when "off".

For the video service, the arrival rate is also a Poisson process, but session duration has a mean of $1/m = 180$ seconds. Video sources generate VBR traffic with an average rate of 384 Kbps or 64 Kbps (fixed for a given topology) and a peak to mean ratio of 3, following the model proposed in [22].

Note that for both services, no actual traffic was generated in the simulations. Video sessions and voice calls were generated simply for measuring transmission rates and performing traffic estimations.

6.1.2 Traffic Prediction

For the sake of simplicity, the resource estimation matrix (Figure 3.11) was generated only using traffic prediction, i.e., the other factors presented in section 3.5.1 were not considered. A Local Gaussian Predictor was used considering that when the number of individual user service flows gets large, the aggregate arrival rate tends to have a Gaussian distribution under the Central Limit Theorem. It is also simple and works well for the purpose of this evaluation, which goal is not to evaluate traffic prediction models, but to concentrate on the evaluation of negotiation models instead. It is described as "local" because only the traffic samples collected at the last measurement interval, called T_{meas}, are used in the computations.

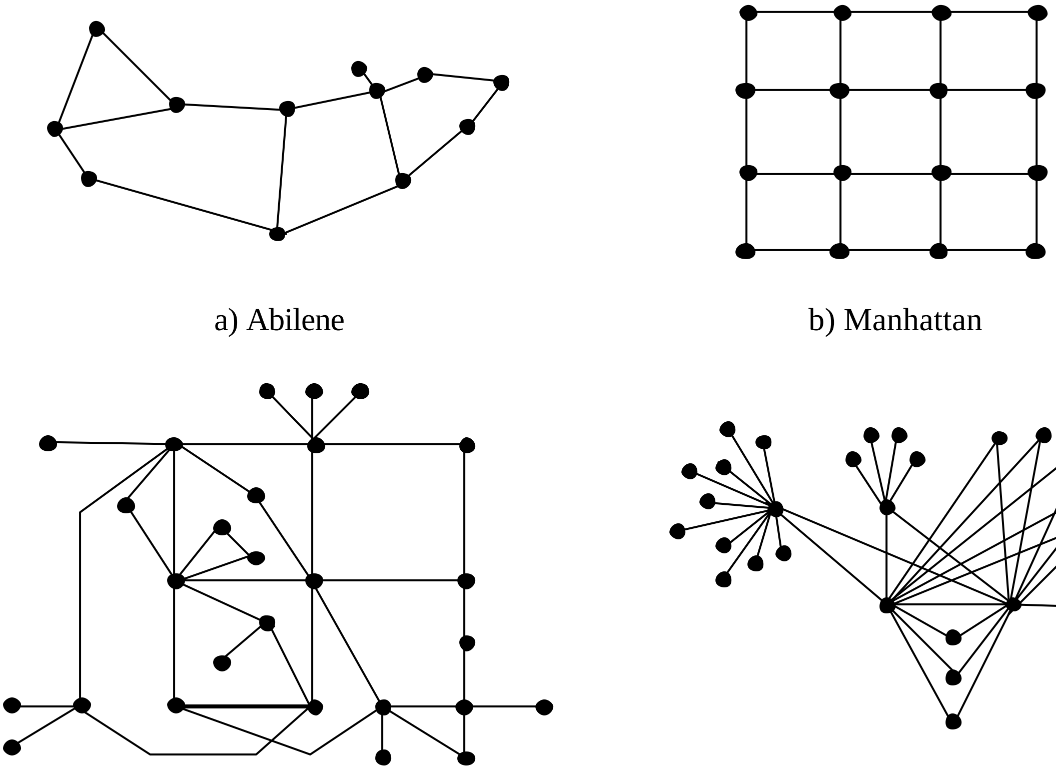
[31] In further simulations, a heavy-tailed Pareto distribution with $a = 1.15$ (where a is the shape parameter) was used for call duration. The results obtained with this traffic model did not significantly differ from those with exponential call duration. Therefore, they are not discussed in the subsequent sections.

The monitoring plane collects traffic samples at regularly spaced time intervals of one second and computes the mean $\bar{x}$, and the standard deviation s, which are sent to the service plane.

The estimation of the throughput to be negotiated is calculated by $\hat{T} = \bar{x} + a\, s$, where a is a multiplier that controls the extent to which the predictor accommodates variability in the samples. In a Gaussian approximation to the negotiated capacity it is expected that the throughput $\hat{T}$ be exceeded with probability $1 - G(a)$, where G is cumulative distribution of the standard normal distribution. In the simulations presented in this chapter, 2.5 was used as the value for a, which yields the low probability of exceeding the predicted throughput of 0.006.

6.1.3 Topologies

The choice of a network topology for simulation is an important step and it should represent as close as possible a network operating under real conditions. For the purpose of this study, four topologies were chosen (Figure 6.1): Abilene [208], GÉANT [90] and RNP2 [170] which are the backbones of the Internet2, European and Brazilian research networks, respectively. The fourth topology is a simple Manhattan network.



a) Abilene

b) Manhattan

c) GÉANT

d) RNP2

Figure 6.1 – Simulated Topologies

The four topologies have interesting features to be analyzed. Abilene (Figure 6.1a) has only 12 domains (GigaPoPs) connected through links with identical capacity of 2.5 Gbps. Manhattan (Figure 6.1b) is also a simple and regular topology, with 16 domains connected through 45 Mbps links. In GÉANT (Figure 6.1c), 26 domains are interconnected in a more complex topology, with links of different capacities, varying from 34 Mbps to 10 Gbps. RNP2 (Figure 6.1d) also has a complex topology, with lower capacity links, varying from 1 Mbps to 25 Mbps. In summary, two simple topologies, which links have the same capacity and two complex topologies with diverse link capacity are considered. From a second classification, two topologies have high capacity links whereas two others have relative low capacity links. As far as the delay is concerned, interdomain links and domains (representing one single path – section 3.5.2) were configured according to a rough approximation of the physical distances, in such a way that requests are not denied for violating it. This information was only used to help evaluating QoS Routing strategies in section 6.3.3.

System load was evenly distributed among domains in Abilene and Manhattan, unlike GÉANT and RNP2, where it was distributed according to the sum of the capacities of the links connected to the domains. When a voice call or video session is generated, a destination for it has to be chosen, in addition to its duration. In Abilene and Manhattan, destination domains were chosen uniformly among the remaining domains. In GÉANT and RNP2 the probability that a domain is chosen as a destination is relative to its in/out capacity (the same criterion for distributing system load). These decisions are motivated by the fact that GÉANT and RNP2 have links with varied capacities, meaning that the amount of in and out traffic is also uneven.

Each topology is modeled as a CDG where each node represents a domain. This is close to reality, since in Abilene each node is a GigaPoP that concentrates many networks, in GÉANT each node is a research network of a different country and in RNP2 each node is a state owned PoP that connects universities and research centers and also the state network. In the hierarchical model, only one level of SE was simulated.

6.1.4 Simulation Results

For each evaluated scenario (topology, load, service, negotiation model and number of phases in the hierarchical model), simulations were carried out, with one negotiation occurring at periods of 60 seconds (the same time was used for the measurement window of the Gaussian predictor, except for Figure 6.3). Domains generate voice calls or video sessions, make traffic predictions and submit the resource requests to be negotiated. Our simulations accepted partial resource grants, as described in section 5.3.3.. For each scenario, 100 replications were executed, using the batch means method for transient removal and collecting values of the metric of interest at the end of each replication. This number of replications was chosen because it represents a good and reasonable compromise between computational cost and statistic reliability [127]. Particularly, 100 replications guarantee a minimum precision of ±3% for every simulated scenario. This hypothesis was not rejected by the c^2 goodness of fit test of the 95% confidence interval.

The results presented in this chapter refer to the average of the metric of interest of all replications. For all results, 99.9% confidence intervals were computed, although the vertical bars representing the intervals were only plotted together with the graphs when they were large enough for enhancing visual information.

6.2 Evaluation of the Gaussian Predictor

Simulations in this section evaluate the Gaussian Predictor, regarding the extent to which it can deal with changes in the aggregate throughput, avoiding underestimation and minimizing overestimation. In this section, a predictor of a single domain was considered and traffic was generated by the voice service only.

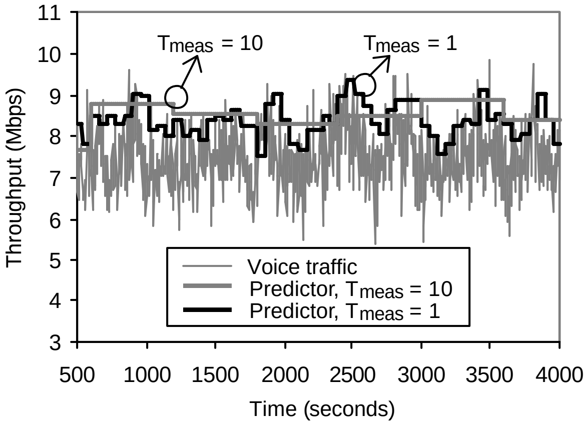


Figure 6.2 – Effectiveness of the Gaussian predictor (r = 240 calls)

The effectiveness of the Gaussian predictor for generating estimates close to the real future traffic is evaluated. Figure 6.2 shows a time series taken from a simulation run during one hour, comparing the traffic generated by the voice service and the predictor estimates for a measurement window T_{meas} of 1 and 10 minutes. It may be observed that with T_{meas}= 1 the predictor is more effective in keeping up with the traffic variation. For T_{meas}= 10 the predictor is not able to adapt to traffic fluctuations at short timescales. In this simulation, the mean squared error (MSE) generated by T_{meas}= 10 was 11% higher than by T_{meas}= 1.

Figure 6.3[32] presents the capacity overestimation for a load r = 240 calls varying the measurement window T_{meas} from 1 up to 30 minutes. The lowest overestimation value (about 13 %) is obtained for T_{meas} = 1 minute, only because it can follow short time-scale traffic fluctuations. For T_{meas} = 20 it presents its highest overestimation value (almost 15 %) and after that up to T_{meas} = 30 it falls down. Thus, it is not strictly necessary to use short prediction measurement windows and consequently intervals between successive negotiation rounds may be more spaced. However, voice traffic volumes typically vary along a day, so that too long measurement windows (more than 1 hour) should be avoided [65]. For all values of T_{meas} simulated, underestimation was below 1 %.

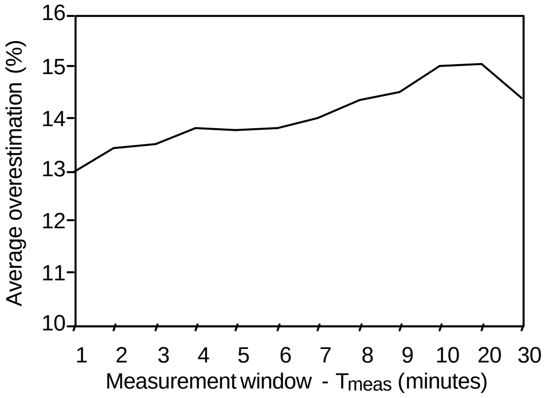
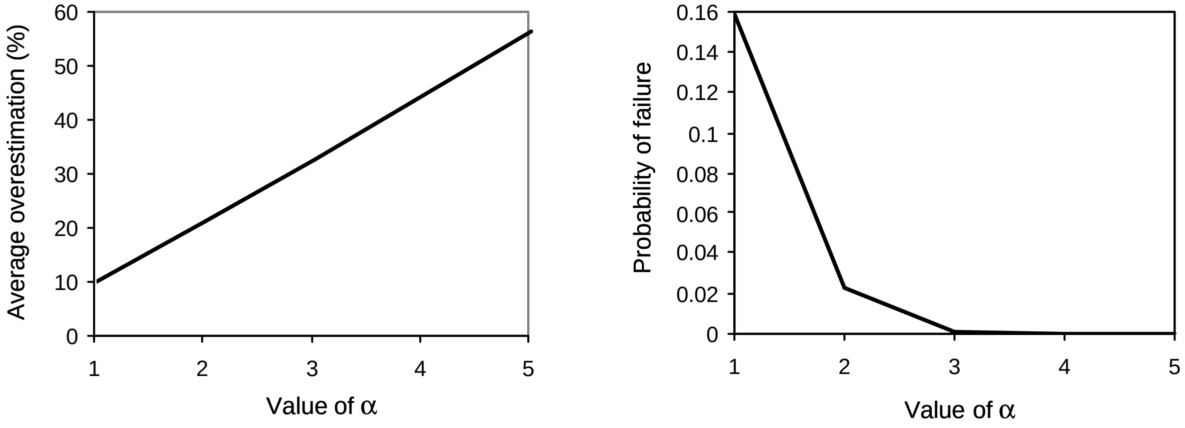


Figure 6.3 – Estimation effectiveness; overestimation versus T_{meas}

The influence of the parameter a on the traffic overestimation is shown in Figure 6.4. The average overestimation increases linearly with the value of a (Figure 6.4a), so that with a = 2.5, which is the value used in these simulations, the overestimation is about 25%. On the other hand, the probability of failure in estimation decreases exponentially with the value of a (Figure 6.4b), so that for a = 2.5 the probability is 0.006. This is the reason behind the choice of 2.5 as the value for a . It is a reasonable compromise between overestimation and estimation failure.

[32] The vertical bars representing the 99.9% confidence intervals are not shown in Figure 6.3, Figure 6.4 and Figure 6.5 because they were not large enough to be visualized.



a) b)

Figure 6.4 - Estimation effectiveness; overestimation versus parameter *a*

Figure 6.5 shows a comparison of overestimation and underestimation with T_{meas} = 1 and load *r* varying from 1 to 1200 calls. For *r* less than 10 calls, estimates are incorrect. Overestimation had a peak of 80% for *r* = 10 and the underestimation was 40 % for *r* = 1. Below *r* = 100 calls, overestimation was higher than 20 %, because for lower loads the aggregate arrival rate does not have a normal distribution. In general, the higher the load, the lower the over and underestimation. In the following sections, a measurement window of T_{meas} = 1 minute was used.

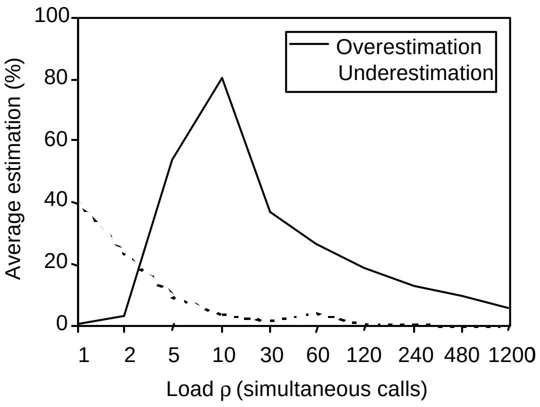


Figure 6.5 – Overestimation and underestimation

6.3 Efficiency

Efficiency in service negotiation refers to the extent to which a negotiation model is able to allocate existing available resources. It is related to both the resource granting for buyer domains and the resource provisioning to seller domains. Despite the current wide supply of raw optic fibers, capacity for exchanging data in the Internet remains always a scarce and expensive resource. Therefore, a negotiation model must be able to allocate as much available resources as possible, whenever there are pending requests. The efficiency in resource allocation is highly dependent on the algorithm used by the negotiation model. Not every negotiation model permits the utilization of the more efficient heuristics for resource allocation, though.

6.3.1 Resource Allocation in Service Negotiation

In order to compare negotiation models according to their efficiency in resource allocation, it is first necessary to state some conditions that must be respected by all models, regardless the styles they adopt (section 5.4).

A CDG is represented as a set of domains interconnected by communication links, and each domain internally by a set of internal paths (the internal paths may also be represented as a single path encompassing the entire domain). The CDG is also characterized by the set of available services (WDSs). Formally, A CDG is represented by $\mathsf{C} = (\mathsf{D}, \mathsf{L}, \mathsf{P}, \mathsf{S})$, where: $\mathsf{D} = \{D_1, D_2, ..., D_d\}$ is the set of domains; $\mathsf{L} = \{L_1, L_2, ..., L_l\}$ is the set of interdomain links; $\mathsf{P}_i = \{P_{i1}, P_{i2}, ..., P_{ip}\}$ is the set of internal paths of domain i ($i = 1,2,...,d$); and $\mathsf{S} = \{S_1, S_2, ..., S_s\}$ is the set of services (WDS instances). For the sake of simplicity, in this evaluation internal paths are represented just as one single path per domain, as $\mathsf{P} = \{P_1, P_2, ..., P_d\}$, that is, $p = 1$.

Each link L and path P can forward traffic from a different sub-set of S, with certain levels of the performance parameters, according to the service offering matrix (Figure 3.13). The negotiation parameter, throughput, is defined for every service. The existence of the definition parameters (delay, jitter and loss) depends on each particular service.

Let R_{ijk} be a resource (throughput) request from source domain i to destination domain j regarding service k, $\forall i, j \leq d$ and $\forall k \leq s$. The value of R_{ijk} is chosen according to the resource estimation matrix (Figure 3.11). Let also G_{ijk} be the resource grant such that $G_{ijk} \leq R_{ijk}$. The request R_{ijk} will only be granted if the following conditions are met:

1. Let L_R and D_R be the set of interdomain links and domains (including i and j) in the end-to-end path.
2. Delay, jitter and loss (in case they apply to service S) must be satisfied in the end-to-end path. More specifically, for both delay and jitter, $\sum_{u \in L_R} d_{Lu} + \sum_{w \in D_R} d_{Pw} \leq d_S$, where d_S is the value defined to service S; d_L and d_P are the values (delay or jitter) currently offered for service S in link L and domain D, respectively. For packet loss, $\prod_{u \in L_R} j_{Lu} \times \prod_{w \in D_R} j_{Dw} \leq j_S$. These two constraints reflect the fact that delay and jitter are additive parameters whereas packet loss is a multiplicative one, as explained in section 5.3.3.
3. For the throughput, $t_{Lu} \geq Min_{ijk}, \forall u \in L_R$ and $t_{Pw} \geq Min_{ijk}, \forall w \in D_R$, where t_L and t_P represent the currently available capacity of link L and domain D and Min_{ijk} is the lowest amount of resources to be granted to a request of service S, such that $0 \leq Min_{ijk} \leq R_{ijk}$.

Independently of whether the negotiation is distributed or centralized, the entity in charge of the resource allocation always tries to allocate the highest amount of resources to a request, observing the following constraints:

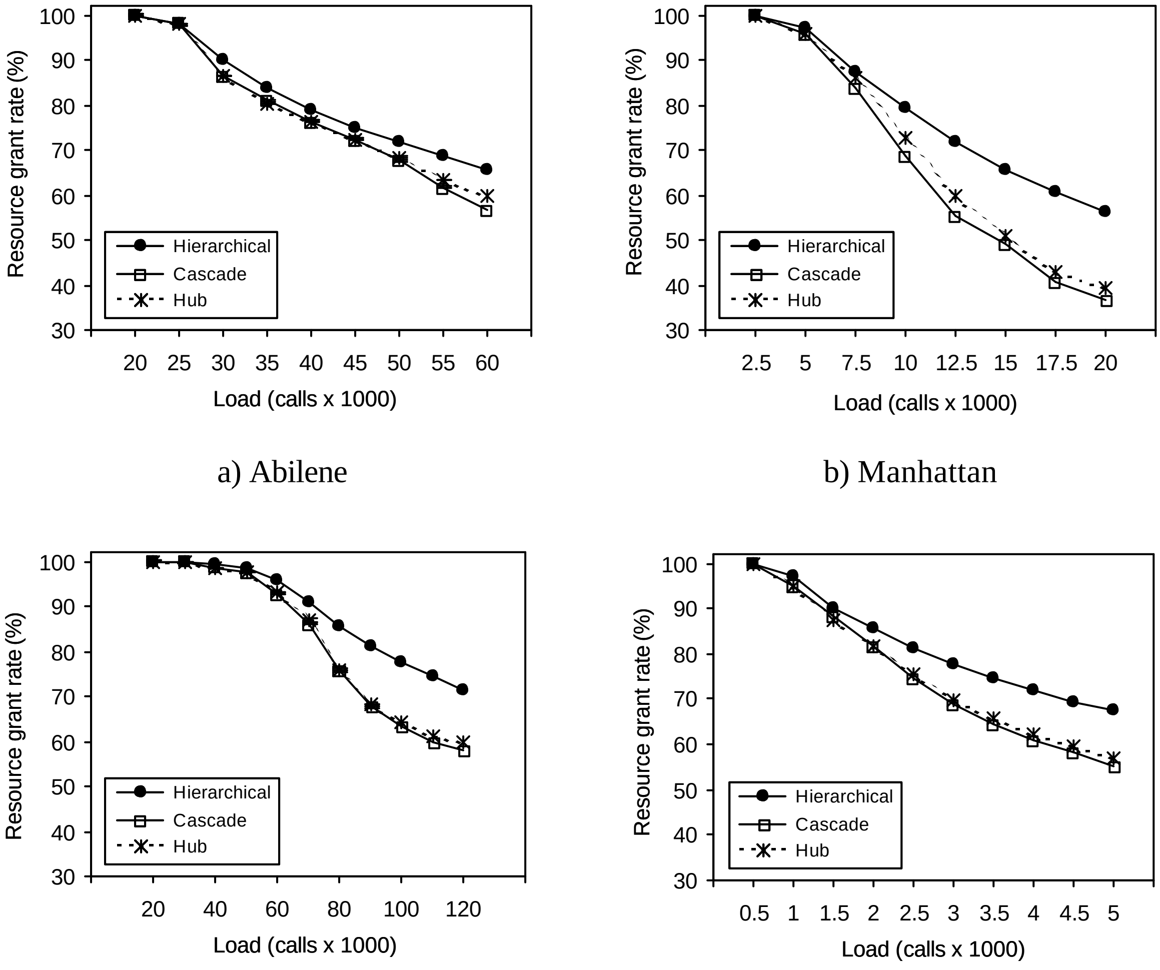
1. $\sum_{r \in R_L} G_r \leq C_L$, $\forall L \in \mathsf{L}$, where R_L is the set of all allocated requests that traverses link L, G_r is the resource grant corresponding to each request, and C_L being the reserved capacity (in other words, the offered throughput) in link L for service S.
2. $\sum_{s \in R_P} G_s \leq C_P$, $\forall P \in \mathsf{P}$, where R_P is the set of all allocated requests that traverses the internal path P, G_s is the resource grant corresponding to each request, and C_P is the reserved capacity in path P for service S.

6.3.2 Resource Grant Rate

The aim of this section is to compare through a simulation study, the cascade, hub and hierarchical negotiation models, using resource grant as an index for efficiency. Let R_i and G_i be respectively the total amount of resources requested by and granted to domain i ($i = 1,2,...,n$) to all destination domains and related to all services. The global resource grant rate is defined as $GR = 100 \times 1/n \sum_{i=1}^{n} G_i / R_i$. The evaluation presented below refers to the average of GR for the 100 replications of the simulation. In this section, the graphs do not show the bars for the 99.9% confidence intervals, due to the low variation of the values collected for GR and consequent small intervals to visualize.

Figure 6.6 shows quite similar results for the four simulated topologies, when the voice user service is considered. The cascade and hub models generated values of GR very close to each other, because or their distributed negotiation style. Since there is no central entity for coordinating the resource allocation, and messages have to be sent to the various domains that take part in the end-to-end path for a given service, one negotiation request may easily interfere with the other ones. This happens because until a domain receives a confirmation message, it has to keep resources pre-allocated. Under light loads (20000, 2500, 20000 and 500 simultaneous calls for Abilene, Manhattan, GÉANT and RNP2, respectively), the GR is 100% for the three negotiation models. In other words, when there is no contention for resources, any method for resource allocation can be used. As the load increases (and the contention as well), it turns out more and more that pre-allocated resources are not integrally confirmed. While resources are pre-allocated, they cannot be allocated to any other request, thus creating a situation where existing resources remain unallocated even though they could be allocated to requests that were not totally fulfilled.

In the hierarchical model, the negotiation is centralized and the SE is responsible for resource allocation. Consequently, the problem of cross-interference does not affect the hierarchical model and hence can allocate more available resources than the other two models. As such, more efficient algorithms can be used (section 5.7.4), based on heuristics developed in the field of graph theory. The results from this section were obtained for a routing passive style and using the *default* shortest-path routing algorithm. In general, the higher the load the better the results of the hierarchical model compared to the cascade and hub models.



a) Abilene

b) Manhattan

c) GÉANT

d) RNP2

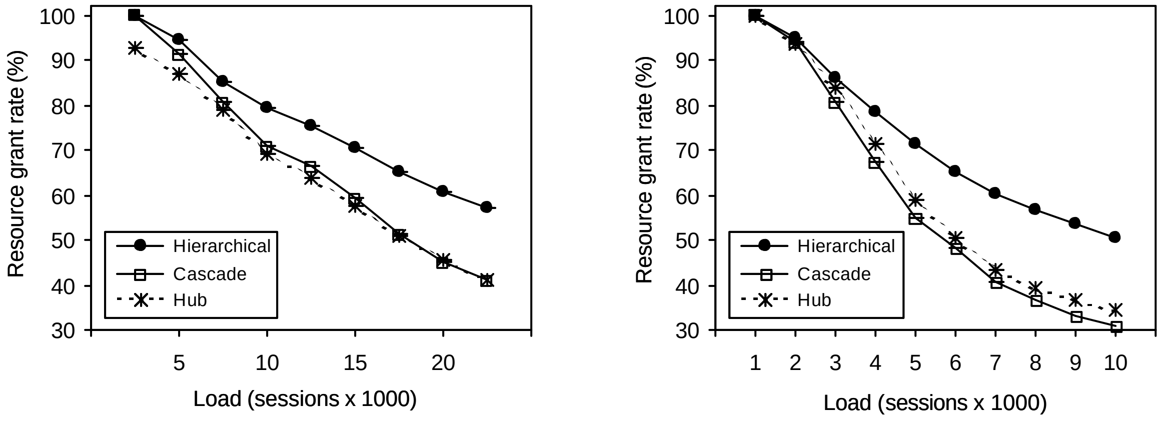
Figure 6.6 – Resource grant rate (*GR*) – voice traffic

Table 6.1 summarizes the gain in resource allocation of the hierarchical model compared to the cascade model, for both high and medium simulated loads. It shows the gain in the *GR* and the amount of increase it represents as a percentage of the *GR* obtained for the cascade model. The highest *GR* gain was 20 for Manhattan, which represents an increase of 49.4%. For medium load, Manhattan had also its biggest *GR* gain, of 10.8, representing 15.8%. These results show that, for the simulated scenarios, there is an incentive for using the hierarchical model, even without using the most efficient algorithms (which are only possible when the routing active negotiation style is used in the CDG).

Table 6.1 – Allocation gain of the hierarchical model

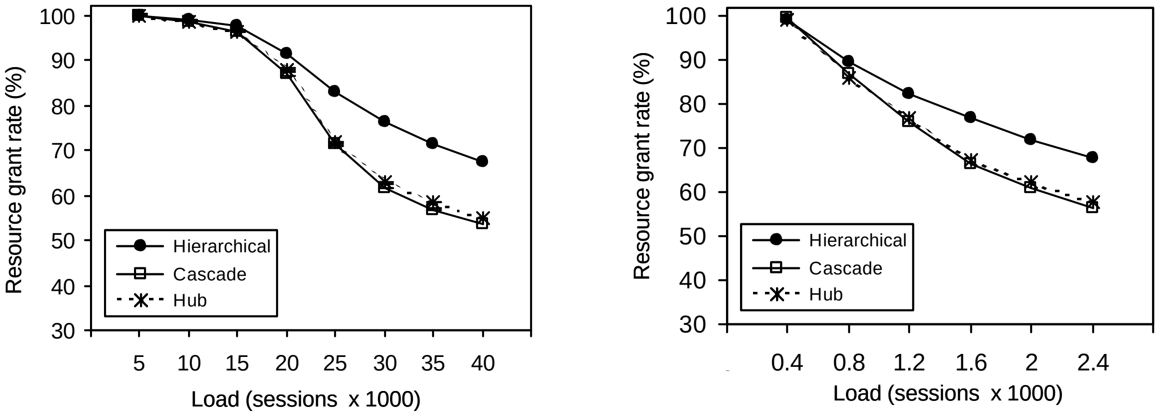
Topology	High load		Medium load	
	Gain (GR)	Increase (%)	Gain (GR)	Increase (%)
Abilene	8.8	15.6	2.6	3.3
Manhattan	20.0	49.4	10.8	15.8
GÉANT	14.5	24.2	4.8	5.6
RNP2	12.0	21.7	9.0	13.1

The same simulations were carried out for the video service. Figure 6.7 shows that there were no significant differences between voice and video services for the simulated scenarios. The curves present the same behaviour as the load increases, with a single exception: the hub model for Abilene was not able to grant 100% of the requested resources for 2500 calls, unlike the other two models. In most cases, the *GR* gain of the hierarchical model was higher than the others for the voice service, reaching up to 20.3 over the Manhattan topology, which also represents an increase of more than 60% in relation to the cascade model.



a) Abilene

b) Manhattan



c) GÉANT

d) RNP2

Figure 6.7 - Resource grant rate (GR) – video traffic

6.3.3 Optimizations of the Hierarchical Model

The results presented in section 6.3.2 consider the hierarchical model operating in the routing passive negotiation style, i.e., it follows the routing information gathered and distributed by the BGP protocol. The goal was to be fair in the comparisons, since the cascade and hub models cannot operate in the active routing style. In this section, the resource allocation of the hierarchical model is further enhanced with optimizations based on techniques similar to those ones used for QoS Routing. The cascade and hub models cannot operate on the routing active style, although they could rely on existing interdomain QoS Routing techniques for optimizing the discovery of adequate paths for service requests [53][54]. In any case, they are bounded to use the shortest path algorithm, since they operate in a distributed negotiation style. When operating in the routing active style, the hierarchical model can use a variety of efficient heuristics for choosing the best possible allocations, as stated in section 5.7.4

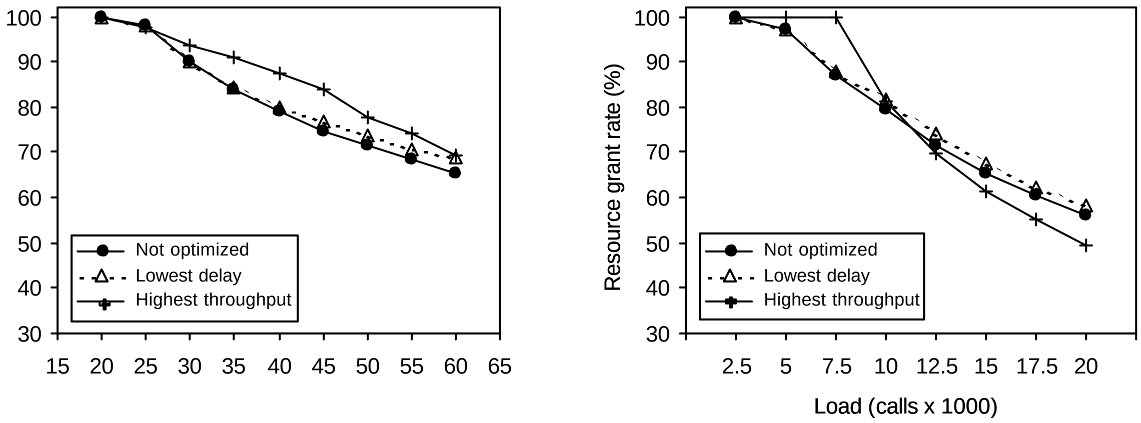
Two different optimization schemes for the hierarchical model are evaluated in this section, based on modifications performed on the basic Dijkstra´s shortest path algorithm. When two or more paths are able to satisfy a request, both schemes have the same behaviour: they select the first path. When the request cannot be fulfilled by any path, they use different heuristics for finding a feasible path:

1. Highest throughput: the chosen path is always the one that has the higher throughput (available resources); where there are more than one path with the same throughput, it chooses the one with the lower delay.

2. Lowest delay: the chosen path is the one with the lowest end-to-end delay; where there are more than one path with the same delay, it chooses the one with the higher throughput.

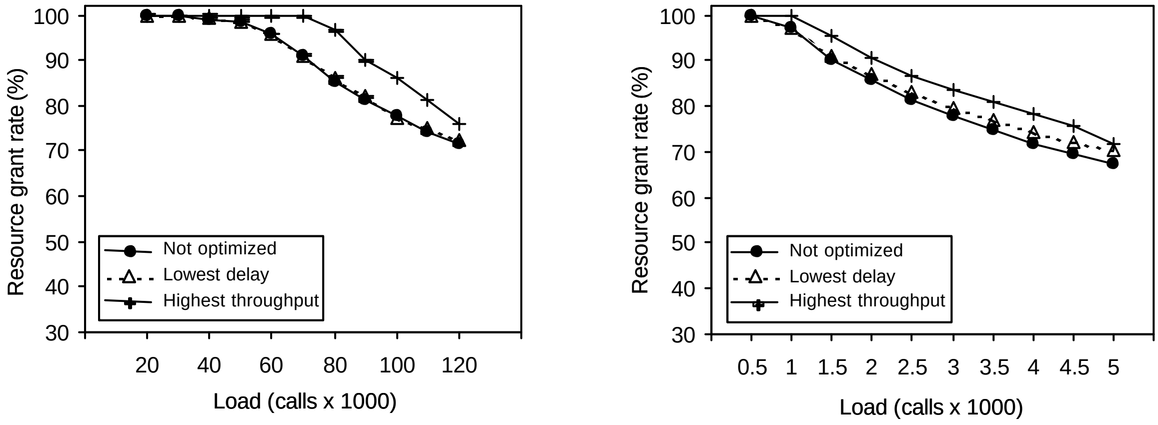
Computational complexity is a concern here, but it does not invalidate the use of such schemes, since there are efficient polynomial algorithms for QoSR when throughput and another QoS parameters (e.g., delay) are considered [220]. Each service request involves a search for an available path for the tuple *<source; destination; service>*. When the SE finishes this process, it informs the domains about requests that were granted, provisioning that must be made and routes that have to be propagated to the routing protocols.

Figure 6.8 shows the simulation results (the GR) of the voice service for the three algorithms of the hierarchical model: the default and not optimized case, based on the routing passive style and the two optimizations described before. Some conclusions can be taken from these simulated scenarios. First, in most cases, the highest throughput heuristic is the more efficient, being able to grant more resources than the other two algorithms. One exception is in Manhattan, from 10000 calls on, where the heuristic fails and the curve falls below the other two ones. Second, the lowest delay heuristic was not able to obtain any significant improvement, so that its additional complexity could be justified. Third, the benefits of deploying a more sophisticated algorithm are always observed for low to medium load. As the load increases reaching its maximum, the improvements are less significant (or they can even become negative).



a) Abilene

b) Manhattan



c) GÉANT

d) RNP2

Figure 6.8 – Optimizations of the hierarchical model – voice traffic

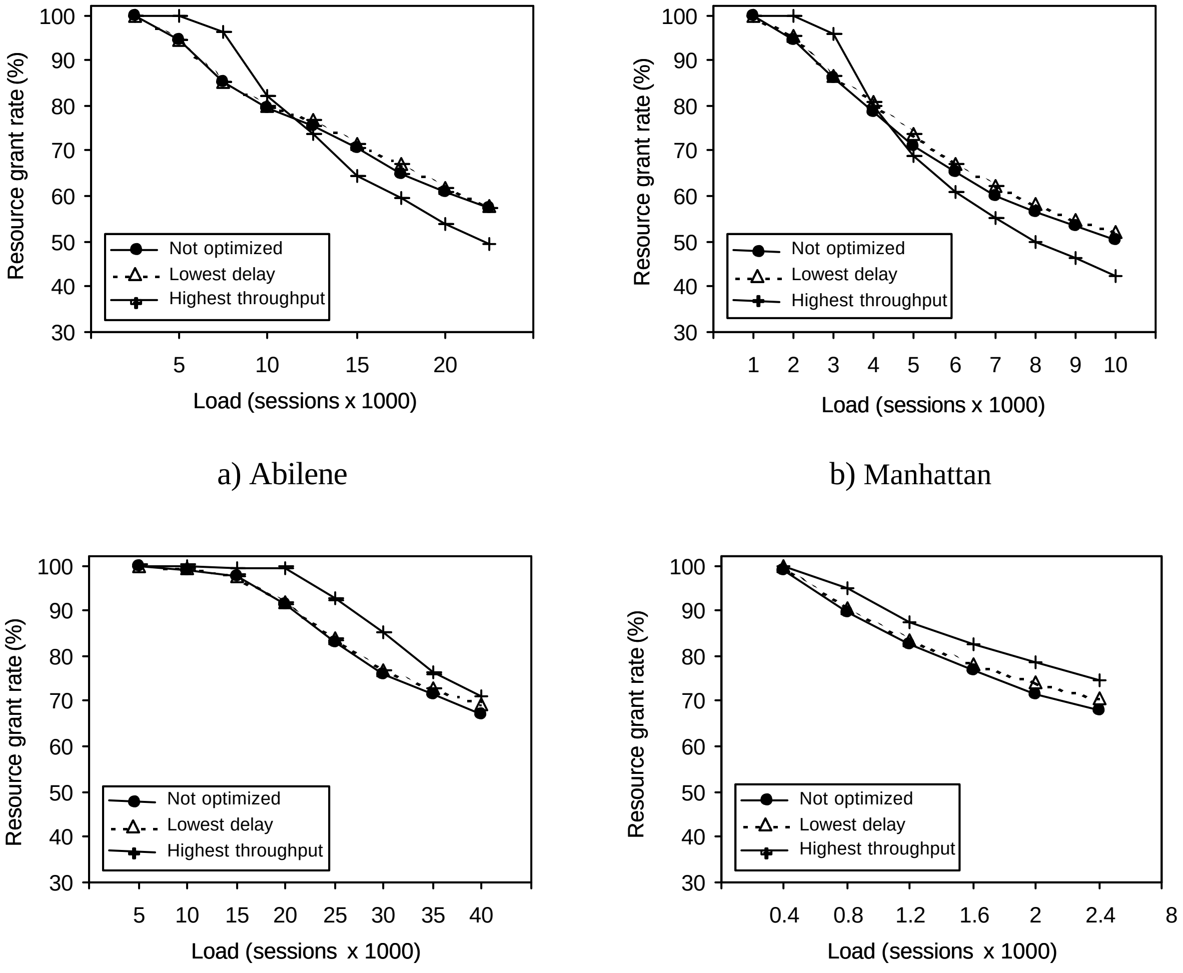
Table 6.2 summarizes the best and worst results of Figure 6.8 giving the GR gain of the highest throughput and the lowest delay heuristics in relation to the unoptimized scenario. In Manhattan, the GR gain of the highest throughput heuristic achieved its best improvement, of 12.4, which represents an increase of 14.2%. Manhattan also had the worst result, corresponding to a decrease of 6.9 in the GR.

Table 6.2 – Optimizations of the hierarchical model - GR gain

Topology	Best result		Worst result	
	Highest Throughput	Lowest Delay	Highest Throughput	Lowest Delay
Abilene	8.9	3.4	- 0.4	- 0.1
Manhattan	12.4	2.7	- 6.9	0.0
GÉANT	11.1	1.1	0.0	- 0.2
RNP2	6.5	3.0	0.0	0.0

The relevance of this information becomes more obvious when combined with the results of section 6.3.2. The hierarchical model without any optimization is able to obtain significant allocation gains over the cascade and hub models, which can be further enhanced with optimization based on less complex heuristics. In general, the highest throughput heuristic was able to grant between 10 and 20 % more resources than the cascade model with low load. The fact that the highest gain is associated to low load scenarios is important, as it is not expected that domains will request much more resources than the network is able to provide. In addition, in order to grant about 100% or the requested resources to all domains, the level of over-provisioning using the hierarchical model is less than that of the cascade and hub models, because it is able to find and allocate more existing resources, which would be left unused otherwise.

The results for the video services (Figure 6.9) show once more that there are no significant differences when the traffic patterns and volumes change. The single remarkable difference is in Abilene, where the highest throughput heuristic fails from 10000 sessions on. This only comes to corroborate with the previous conclusion that the optimizations are better for relatively low loads, which is the intended scenario of the Chameleon architecture.



a) Abilene

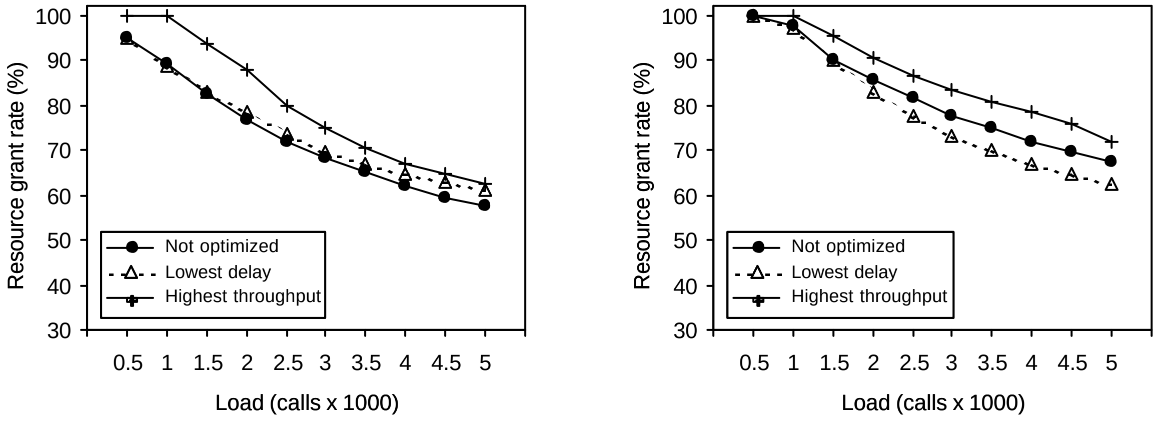
b) Manhattan

c) GÉANT

d) RNP2

Figure 6.9 – Optimizations of the hierarchical model – video traffic

In order to compare the optimizations of the hierarchical model under varied conditions, additional simulations were carried out with slightly different configurations. Figure 6.10 shows the results for two changes in the RNP2 topology. The first change (Figure 6.10a) was to decrease the capacity of the main interdomain link (from Rio de Janeiro to São Paulo) from 20 Mbps to 1 Mbps. It can be observed that the highest throughput heuristic was able to find available resources in alternative paths, so that the problems caused by the lack of resources could be hidden as much as possible from the requesting domains. On the other hand, the other two algorithms were not able to grant 100% or the requested resources even for the lowest load (500 calls). The basic algorithm (not optimized) must follow the normal interdomain routes, since it operates in the routing passive mode. As such, it is not able to discover alternative paths. The lowest delay heuristic tries to allocate requests from the interconnected domains through the changed link, since it has the lowest end-to-end delay.



a) b)

Figure 6.10 – Configurations changed for RNP2; a) main interdomain link with lower capacity; b) main interdomain link with higher delay

The second change to RNP2 was to increase the delay of the same link from 5 ms to 50 ms. It is shown in Figure 6.10b that the basic algorithm and the highest throughput heuristic were not influenced by this change, since it did not cause a service denial, as the maximum delay of the service was not exceeded. On the other hand, the lowest delay heuristic had a bad performance, with a grant rate even worst than the not optimized case. The reason for this is straightforward: it chooses alternative (and longer) paths, which have a lower delay now. Thus, a larger amount of resources is consumed to deal with the same number of service requests and the outcome is a decrease in the resource grant rate. The same problem happened with the highest throughput heuristic in Figure 6.8b, Figure 6.9a and Figure 6.9b, where the index *GR* falls below the other two models due to "bad allocations". This result is in line with the well-known preference in the network community for routing algorithms based on the shortest paths (considering the number of hops), which consumes fewer resources. An interesting aspect is that performance is not changed for the load up to 1500 calls. This is because the changed link does not represent a bottleneck with this load.

Similar results were obtained with the other three topologies and the video service, although they are not presented here. The results of changing the configurations are useful at the extent they support the statement that heuristics based on the throughput are more effective than heuristics based on other performance parameters, such as the delay.

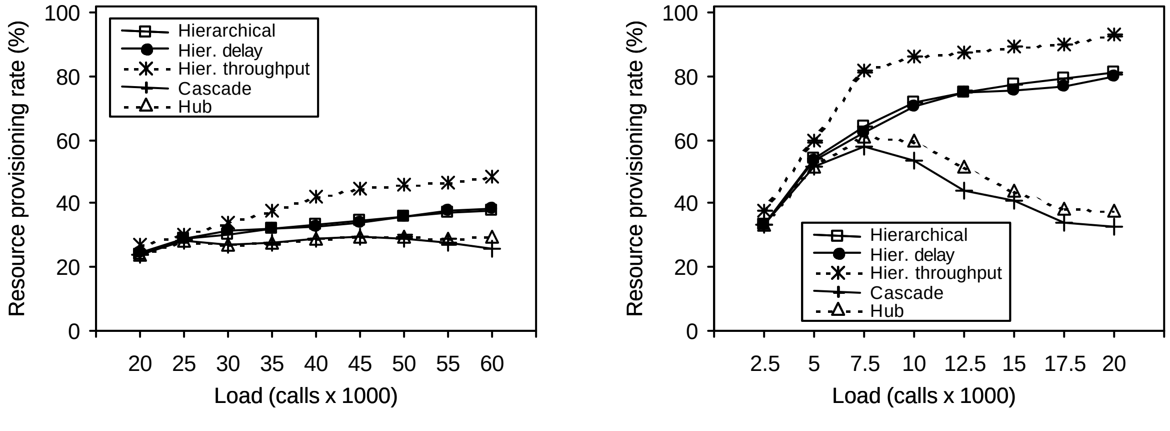
6.3.4 Resource Provisioning Rate

In this section, the goal is to compare the negotiation models through an index that shows the degree of provisioning of the interdomain links as a result of the negotiation process. In the last sections, the evaluation focused the standpoint of service buyers, whereas in this section, it focuses the standpoint of service sellers. Particularly, the index reflects the percentage of the resources offered by seller domains that were sold as a result of the resource allocation and then effectively provisioned. Let O_i be the quantity of resources offered by domain i ($i = 1,2,...,n$) including all interdomain links and S_i the resource really sold by the negotiation. Then, the resource provisioning rate is defined as $PR = 100 \times 1/n \sum_{i=1}^{n} S_i / O_i$. The results presented in this section refer to the average of PR for the 100 replications of the simulation. As in the previous sections, a 99.9% confidence interval was used, but the bars are not shown because they are not significant.

Figure 6.11 depicts the results of the PR for the voice service, considering the three versions of the hierarchical model (not optimized, highest throughput and lowest delay), and the cascade and the hub models. It can be observed that under the highest throughput optimization for the hierarchical model the index PR is always higher than under the other models. In other words, it is able to make a better use of the existing resources than the other models, which is very useful for doing interdomain traffic engineering. In the best case, for Manhattan, PR of the highest throughput heuristic is 48 points (145%) and 60 points (181%) higher than the basic hierarchical and the cascade model, respectively. The basic hierarchical and the lowest delay models had very similar results. The same happened to the cascade and hub models. Another remarkable point is that even the basic hierarchical model (operating in the routing passive style) is able to yield much higher provisioning rates than the cascade and hub models. The results of resource provisioning are in general much more significant than those of resource grant in quantifying the efficiency of the hierarchical model.

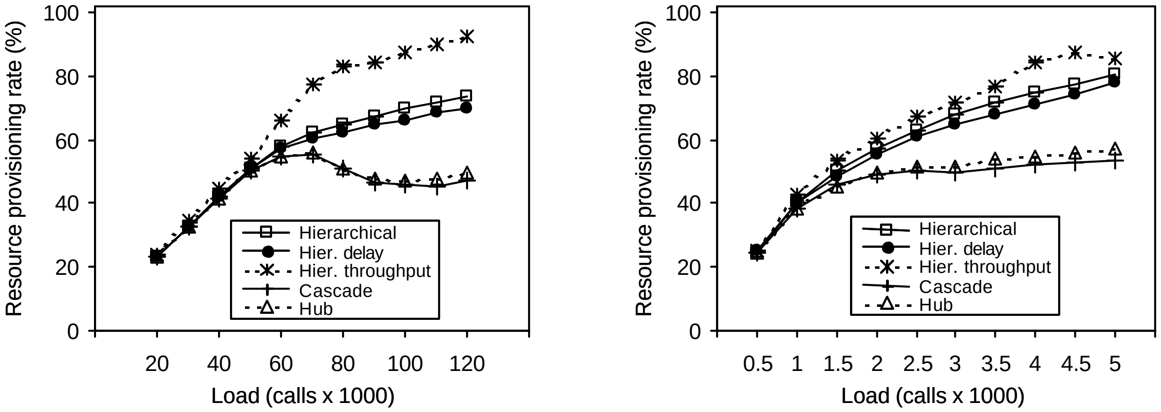
The relevance of the results of Figure 6.11 becomes more evident when they are compared to the results of resource grant in Figure 6.6 and Figure 6.8. Two important observations can be made. Firstly, there is an obvious relationship between resource grant and provisioning: the higher the grant of a negotiation model, the higher the provisioning it produces. Secondly, as the load increases, the grant rate decreases (due to lack of resources) and the provisioning rate increases (because the nominal resource grant also suffers an increase). Both conclusions fail however under certain conditions, as explained in the following paragraphs.

Figure 6.8b shows that the highest throughput model is more efficient with light load but falls below the other basic hierarchical and the lowest delay models from 10,000 calls on. However, in Figure 6.11b the curve for the provisioning of this model remains higher than that of the other two models. What apparently is incoherent can be explained by the conclusions of section 6.3.3 with relation to the results of Figure 6.10. In this case, the highest throughput model found longer paths, so that although it generated a higher provisioning rate, the grant rate was lower. The reason is that more resources are provisioned to some requests, that otherwise could be allocated to others. Therefore, in this scenario a higher provisioning rate do not represent a higher resource grant rate.



a) Abilene

b) Manhattan



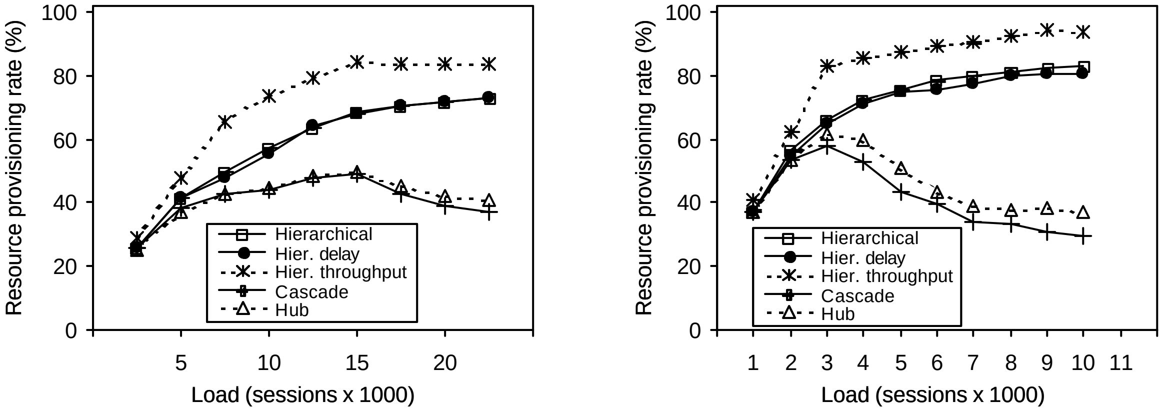
c) GÉANT

d) RNP2

Figure 6.11 – Resource provisioning rate (PR) – voice traffic

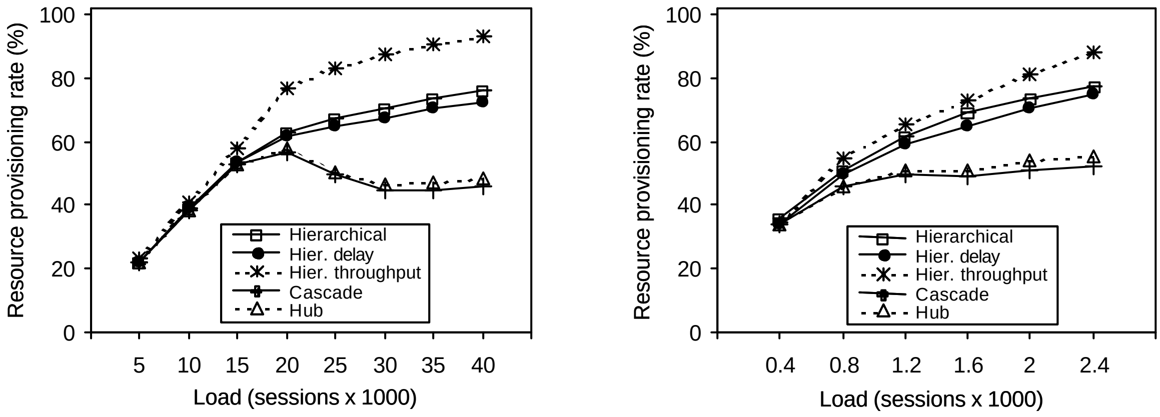
The second conclusion fails for the cascade and the hub models. Because of the cross-interference among different requests, some of them are not granted so many resources as the topology supports. This situation gets worse as the load increases, to a point that in some cases the slope of the curve changes its direction and starts to fall (Figure 6.11b and c). This is the explanation for the reason why the basic hierarchical model to become more efficient in granting resources than the cascade and the hub models, although the three models use the same heuristic (the shortest path).

Similar results were obtained for the video service, as depicted by Figure 6.12. The afore-mentioned problem with the highest throughput model happened again for Manhattan (Figure 6.9b and Figure 6.12b), but also for Abilene (Figure 6.9a and Figure 6.12a). This shows that traffic volumes can influence the results of service negotiation, as long as this information is used by the algorithm of resource allocation (that is the case of the highest throughput model).



a) Abilene

b) Manhattan



c) GÉANT

d) RNP2

Figure 6.12 – Resource provisioning rate (PR) – video traffic

6.4 Fairness

Fairness in service negotiation refers to the degree a negotiation model is able to distribute available resources in the network in such a way not to give preference to some domains to the detriment of others. It is measured by the variation of the resources grant rate for the domains. Fairness only comes into play when there is contention for resources, that is, when the amount of requested resources (by all domains) exceeds the capacity of network paths where traffic is forwarded through. Obviously, not every network resource may be fully used, either because requests cannot be spread over different routes, or simply because resources are not located where they are needed.

Fairness also largely depends on the algorithm used by the negotiation model for resource allocation. The more efficient an algorithm allocates resources, the lower the contention and the higher the fairness in its distribution is. Therefore, in order for the evaluation of fairness not to be biased, and for the sake of comparison the same algorithm was used for the three negotiation models (based on the basic shortest path heuristic). The aim was to isolate the effects of the efficiency and fairness criteria.

6.4.1 The Problem of Unfairness

In theory, it would be interesting if the percentage of granted resources was exactly the same for every domain in each negotiation. However, negotiation models are not able to distribute resources in a fair manner among domains. In general, either some domains are systematically benefited or harmed, or in different moments, resources are unevenly distributed in a random or unbalanced way.

In the cascade and hub models, this happens due to the autonomy of each domain to request resources whenever it is willing to. The decision of reserving resources or not is also taken independently by each domain for each request, disregarding other requests. In the hierarchical model, where the SE performs resource allocation simultaneously for every participant domain, resource grant is strongly influenced by the order in which the domains are chosen. As a rule, domains taken first always get more resources, independently the particular order that is adopted.

6.4.2 Proposed Solution for the Hierarchical Model

Our proposed solution to obtain fairness in negotiation for the hierarchical model is to divide negotiation into p phases ($p \geq 1$). In each phase, the SE tries to allocate a slice equivalent to $1/p$ of the resources requested by each domain. Although discrete, this resource allocation technique tends to behave similarly to continuous models, when a sufficiently large number of phases is used. The objective of this solution is to avoid the allocation order to cause unfairness in resource distribution. The final difference in the percentage of granted resources for each domain must be small, only resulting from characteristics of the topology and the amount of requested resources.

Let R_i be the amount of resources requested by domain $D_i \left(i = 1,2,...,n\right)$ in a given negotiation round and G_i the granted amount. Let $P_i = \left(G_i/R_i\right)\times 100$ be the percentage of granted resources to domain D_i. In each phase, $j \left(j = 1,2,...,p\right)$ the algorithm tries to allocate to domain D_i a resource slice of $R_{ij} = R_i/n$, granting the amount G_{ij}. Let also $P_{ij} = \left(G_{ij}/R_{ij}\right)\times 100$ be the percentage of resources granted to domain D_i in phase j. Thus, the maximum unfairness in each phase j is given by $U_j = \left(\max\left(P_{ij}\right) - \min\left(P_{ij}\right)\right)/p$, where $\max\left(P_{ij}\right)$ and $\min\left(P_{ij}\right)$ correspond to the maximum and minimum resource granted percentages. It is easy to see that when $\max\left(P_{ij}\right) = \min\left(P_{ij}\right)$ then $U_j = 0$ and also that when $p \to \infty$ then $U_j \to 0$. It means that when the maximum and minimum percentages are of equal values, there is no unfairness at all. This is practically impossible though, unless there is no resource contention and $\max\left(P_{ij}\right) = \min\left(P_{ij}\right) = 100/p$. On the other hand, when the number of phases gets sufficiently large, the difference in each phase becomes negligible. As each phase contributes with the allocation of $100/p$ % of the resources, the maximum global unfairness committed in the negotiation is given by $U = \sum_{j=1}^{p} U_j$.

Hence, if in phase j the capacity of a link or domain is exhausted, the maximum difference among percentages of granted resources to domains that depend on that link (or domain) can be $\Delta_j = \max(P_{ij}) - \min(P_{ij})$. If this link (or domain) belongs to the critical path from each domain to every other domain, then no more resources exist to be allocated and the global difference Δ is equivalent to Δ_j . On the other hand, if there are still available resources in other paths to interconnect any two domains (which is more likely to happen), then allocation continues further and the maximum difference is likely to increase.

6.4.3 The Fairness Index

A variety of indexes may be used for evaluating fairness of negotiation models. The aim here is to quantify differences in resource distribution amongst the domains through the variation of percentages of granted resources to each domain. In section 6.4.2, the difference between the maximum and the minimum (i.e., the range) was used for analyzing the solution for the hierarchical model. The range can be used as a dispersion index in this case because the maximum and the minimum values have well defined bounds (they are percentages) [119]. Additionally, other dispersion indexes can be used, such as the standard deviation, the interquantile range (IQR), the range between percentiles (e.g., 10- and 90- percentiles), or the variation coefficient that shows the ratio of the standard deviation to the mean.

A fairness index that includes all samples (i.e., percentages), adapted from Raj Jain ([119], p. 36) is defined as:

$$J = \frac{\left(\sum_{i=1}^{n} P_i\right)^2}{n \sum_{i=1}^{n} P_i^2} \tag{6.1}$$

where P_i is the percentage of granted resources to domain D_i an n is the number of domains. $J = 1$ when every domain obtains the same percentage of resources. As the variation of granted resources increases, the value of J decreases until $1/n$, when a single domain obtains 100% resources and the others obtain no resource at all.

In general, values of J below 0.9 reveal high variations in the distribution of resources among domains, but this is not an accurate representation of the reality and may be even misleading. Values generated by J are concentrated around 1 instead of being uniformly distributed between 0 and 1. This may lead to misunderstandings of the outcomes, since it is normal to think that 0.9 is something close to a desirable result. Therefore, a linear transformation on index J is proposed, in order to improve the common intuition about it, which spreads results between 0 and 10.

$$F_L = \begin{cases} \dfrac{(J-L)}{(1-L)} \times 10 & , if \;\; J > L \\ 0 & , if \;\; J \leq L \end{cases} \tag{6.2}$$

where L is the cut lower bound, so that for every value of J below it, F_L is 0. In this evaluation, the index $F_{0.7}$ will be used as a particular variation of F_L with $L = 0.7$. This lower bound was chosen (after several tests with different bounds), because it provided the best visualization of the results. The interpretation for L is that the closer to 1 its value is, the lower the values generated by F_L. The forthcoming simulations (next sections) are based on $F_{0.7}$, and they present results that are coherent with other dispersion indexes (according to our simulations).

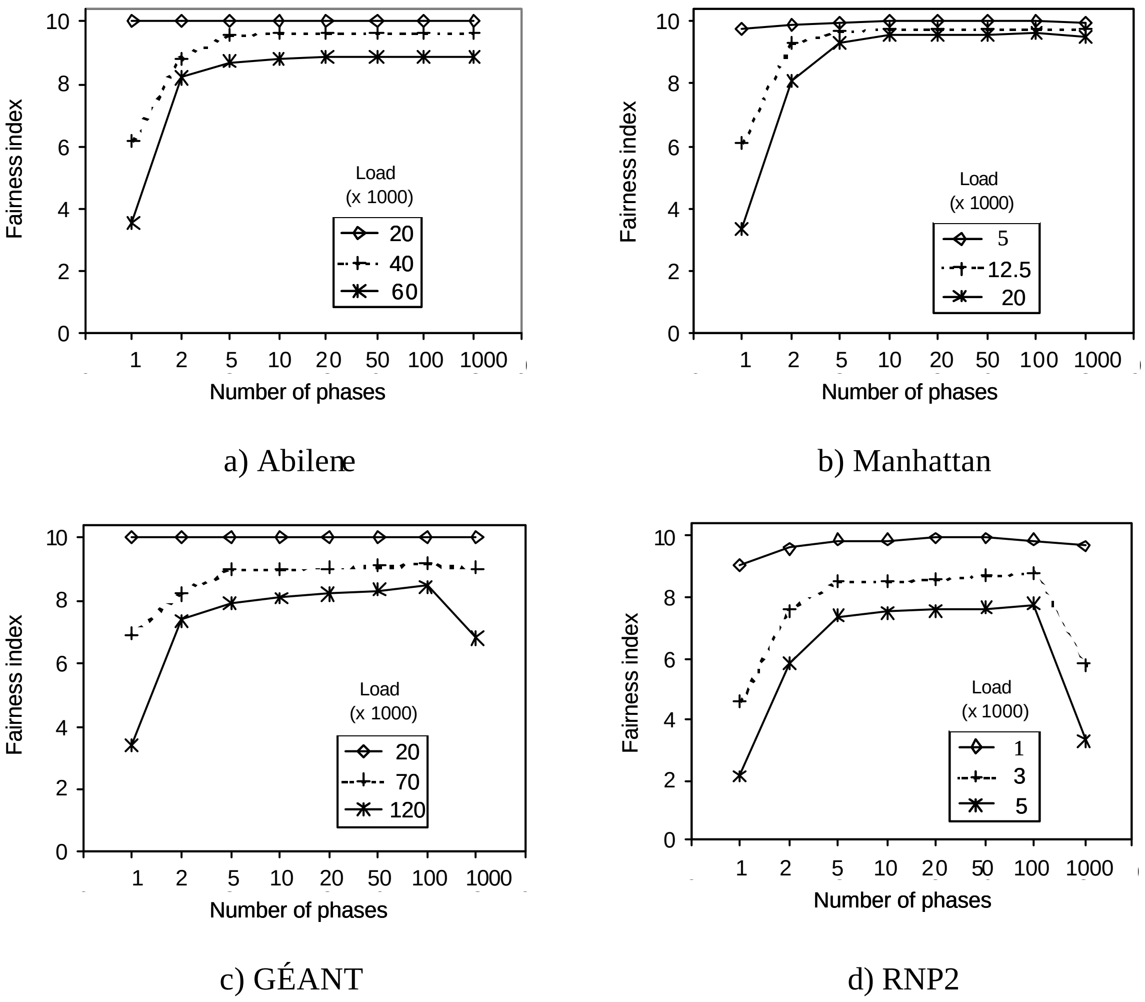
Table 6.3 – Interpretation of the fairness index

Range of $F_{0.7}$	**Fairness in Resource Distribution**
$9 \leq F_{0.7} \leq 10$	Desirable (high fairness)
$7 \leq F_{0.7} < 9$	Acceptable (medium fairness)
$5 \leq F_{0.7} < 7$	Bearable (low fairness)
$0 \leq F_{0.7} < 5$	Unacceptable (unfairness)

Table 6.3 shows a plausible interpretation of $F_{0.7}$ values based on previous experience. It is useful for evaluating simulation results of the next sections.

6.4.4 Comparing Phases in the Hierarchical Model

In section 6.4.2 it was shown that in the hierarchical model, when the number of phases tends to the infinity, the range between the maximum and the minimum grant percentages tends to zero. As the number of phases is directly proportional to the computational cost of the negotiation, the idea is to keep it as low as possible, without harming the fairness. Therefore, in this section the simulations tried to find out the necessary number of phases so that the fairness index $F_{0.7}$ had a steady behaviour. In other words, finding a point by and which the computational cost associated to increasing the number of phases does not compensate the additional improvement in fairness.



a) Abilene

b) Manhattan

c) GÉANT

d) RNP2

Figure 6.13 – Influence of the number of phases (voice service)

Figure 6.13 shows index $F_{0.7}$ along with load curves related to the number of phases, for the four topologies. The numbers of phases simulated were 1, 2, 5, 10, 20, 50, 100 and 1000. The graphs depict that, from 5 phases on, $F_{0.7}$ presents a relatively steady behaviour for all topologies, that is, its increase is not significant. One can also observe that the higher the load is, the lower the value of $F_{0.7}$ is. This can be explained by the increase in resource contention as domains request higher and higher amounts of resources. With only a few phases (one or two), domains selected for first in the allocation order[33] obtain quite a lot of resources whereas the rest obtains very little or nothing. The graphs in Figure 6.13 present only 3 loads (the highest, the lowest and an intermediate), although between 7 and 10 different loads were used for each topology (see Figure 6.14).

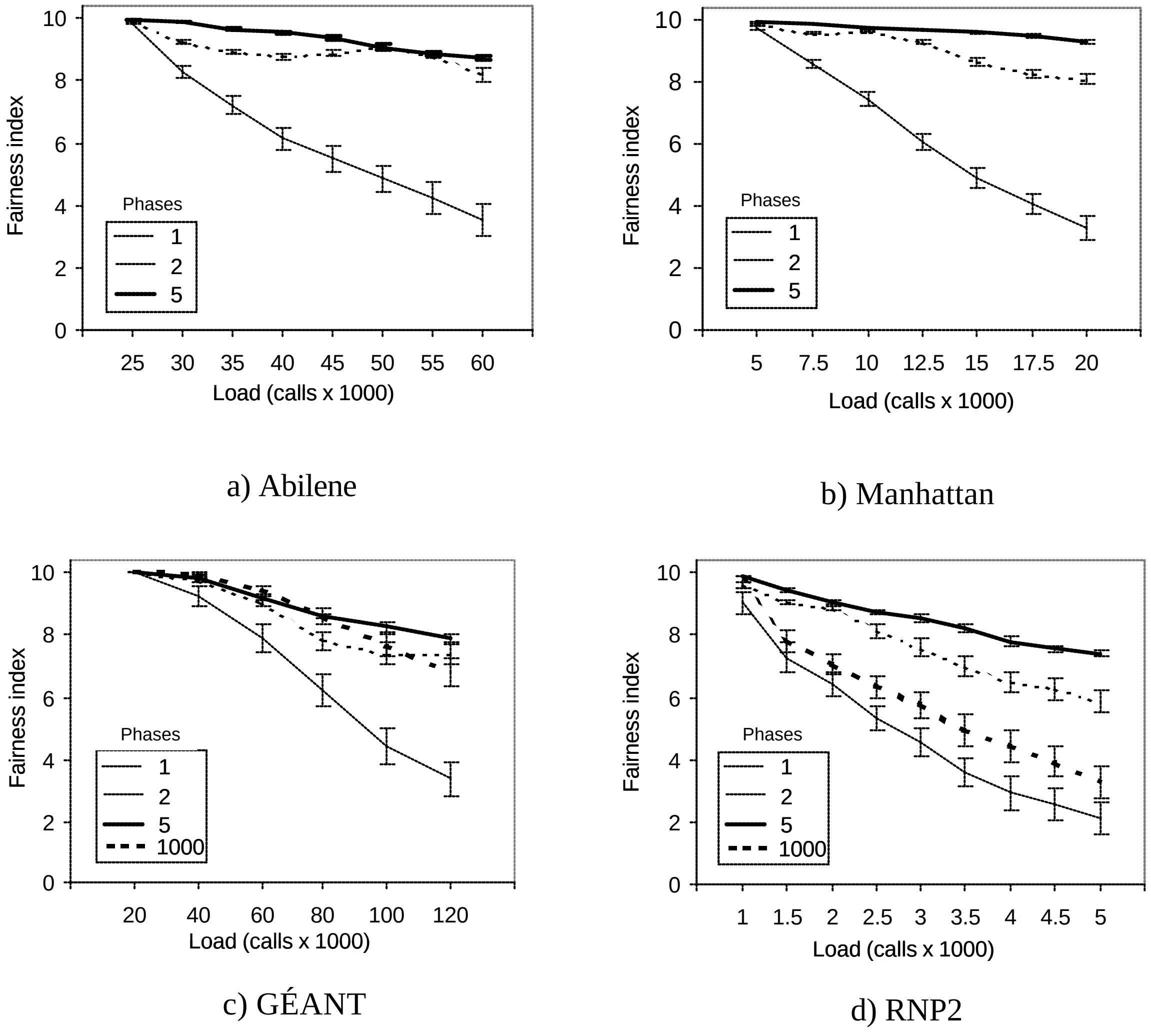
In Abilene (Figure 6.13a), for instance, for 20.000 calls $F_{0.7}$ achieved value 10 in all phases. This means that the network was not overloaded, i.e., there was no resource contention and the network was able to completely fulfill all requests. For 60.000 calls, the network was overloaded and only around 65% of the requested resources were granted (for any number of phases). For one phase, the value of $F_{0.7}$ was 3.52, which indicates a high degree of unfairness, according to Table 6.3. Some additional information corroborates with this conclusion. The average range between maximum and minimum granted percentages was 93.52 (the highest value being 100), the average IQR was 55.87 and the average standard deviation was 31.97. A similar behaviour was observed for Manhattan (Figure 6.13b).

GÉANT (Figure 6.13c) and RNP2 (Figure 6.13d) presented similar behaviours, with higher slopes in the $F_{0.7}$ curves (compared to Abilene and Manhattan) and a violent fall for 1000 phases. At a first glance this result may seem surprising, as one would expect that fairness had a continuous enhancement with a growing number of phases. However, it is due to the fact that the simulator works with integer allocation units of Kbps. The rest of the division of the requested amount by the number of phases is assigned to the first phase. In both topologies the distribution of the load among domains is very uneven and some domains request a small amount of resources. The straightforward consequence is that in some scenarios the behaviour for 1000 phases is similar to that of 1 phase. For example, if a domain requests 900 Kbps, in the first phase the algorithm tries to allocate 900 Kbps and nothing during the remaining phases. Since this is due to a simulator idiosyncrasy, the theoretic conclusion of section 6.4.2 about the asymptotic behaviour of the phase number is not invalidated. Furthermore, from 5 phases on the increase in the computational cost does not compensate the small additional gain in fairness, therefore nullifying the advantage of using 1000 phases in a real case.

Figure 6.14 depicts the same reality. Curves represent phases and bars show 99.9% asymptotic confidence intervals. It can be observed that there is a significant gap between curves of 1, 2 and 5 phases for all topologies. Since often confidence intervals are not overlapped, a logical conclusion is that there is advantage in changing the number of phases

[33] At each phase, the order wherein domains are taken for the allocation process is random, according to a uniform distribution. The reason is not giving preference to any particular domain, since the first domains have an

from 1 to 2 and 5. Beyond 5 phases, curves and confidence intervals are overlapped. As there is no visible distinction among these phases, the graphs do not show the curves for these numbers of phases. For GÉANT (Figure 6.14c) and RNP2 (Figure 6.14d) the curves for 1000 phases has also been included, because of their atypical behaviour.



a) Abilene

b) Manhattan

c) GÉANT

d) RNP2

Figure 6.14 – Gap between phases (voice service)

A conclusion that can be drawn from this evaluation is that, excluding some exceptions of little relevance, the higher the number of phases in the hierarchical model, the higher the fairness index [126]. However, this process implies a higher computational burden as well, which must be reduced as much as possible. From as low as 5 phases, the fairness index approaches its maximum, presenting steady results for a variety of scenarios, with different topologies, loads and services (for the video service, the results were quite similar). Finally, a very important aspect is that the improvement in fairness did not affect the efficiency in

advantage because at the beginning of the allocation there are more available resources.

resource allocation, presented in 6.4.4. The difference in the GR was negligible for all numbers of phases (from 1 up to 1000) in all simulated scenarios.

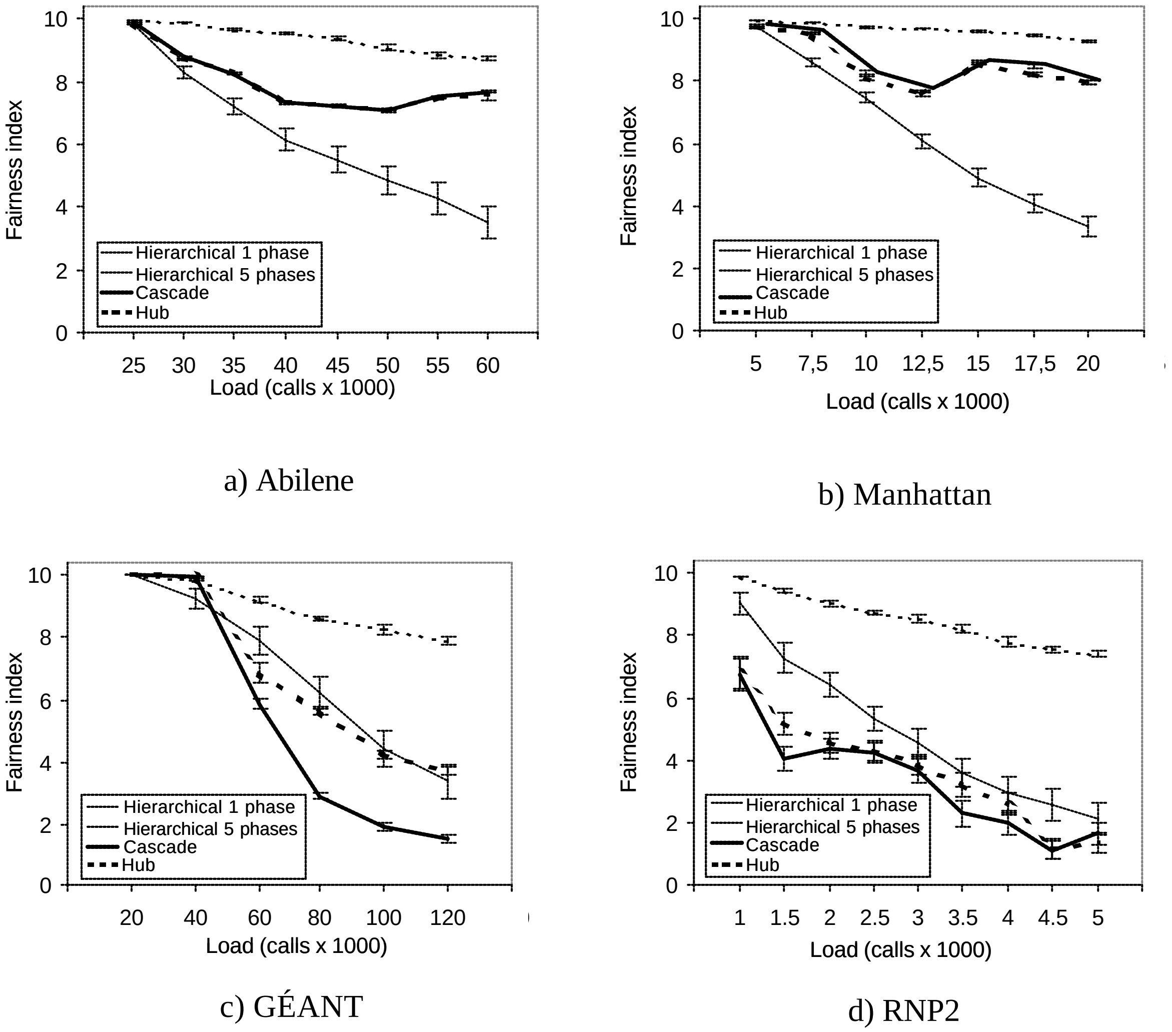
6.4.5 Comparing Negotiation Models

In this section, the goal of the simulations was to compare negotiation models with respect to the fairness criterion. The cascade and hub models are compared to two versions of the hierarchical model: with 1 and 5 phases. These numbers of phases were chosen because 1 phase represents the worst result and, on the other hand, from 5 phases on there were no significant benefits observed (section 6.4.4).

Figure 6.15 shows that the hierarchical model had a steady behaviour in all topologies. The reduction of the value of $F_{0.7}$ is inversely proportional to the load increase in a ratio that does not substantially vary among the evaluated topologies, for both 1 and 5 phases. As the analyzed topologies portray the reality of current networks[34], it shows that the hierarchical model tends to have a better adaptation to different scenarios, according to the fairness criterion.

Unlike the hierarchical model, the cascade and hub models presented huge variations for different topologies. For simpler topologies, Abilene (Figure 6.15a) and Manhattan (Figure 6.15b), results of $F_{0.7}$ are quite similar for both models, being located somewhere between the two cases of the hierarchical model (1 and 5 phases). On the other hand, for GÉANT (Figure 6.15c) and RNP2 (Figure 6.15d) their results were even worse than the hierarchical model with 1 phase. This is because of their varied link capacity and the consequent difference in the domain load and requests. In such a configuration scenario, the domains with lower capacity (that request less resources) suffer more because when the network is overloaded they easily are not granted any resources. As higher capacity domains represent a large portion of the network load, even when the network is overloaded they can obtain reasonable resource percentages. Even though such an allocation is efficient, it may cause unfairness, since the metric being evaluated is the resource grant rate to each domain.

[34] Manhattan topology in unreal, but its regularity and simplicity are very useful to the subject of evaluation.



a) Abilene

b) Manhattan

c) GÉANT

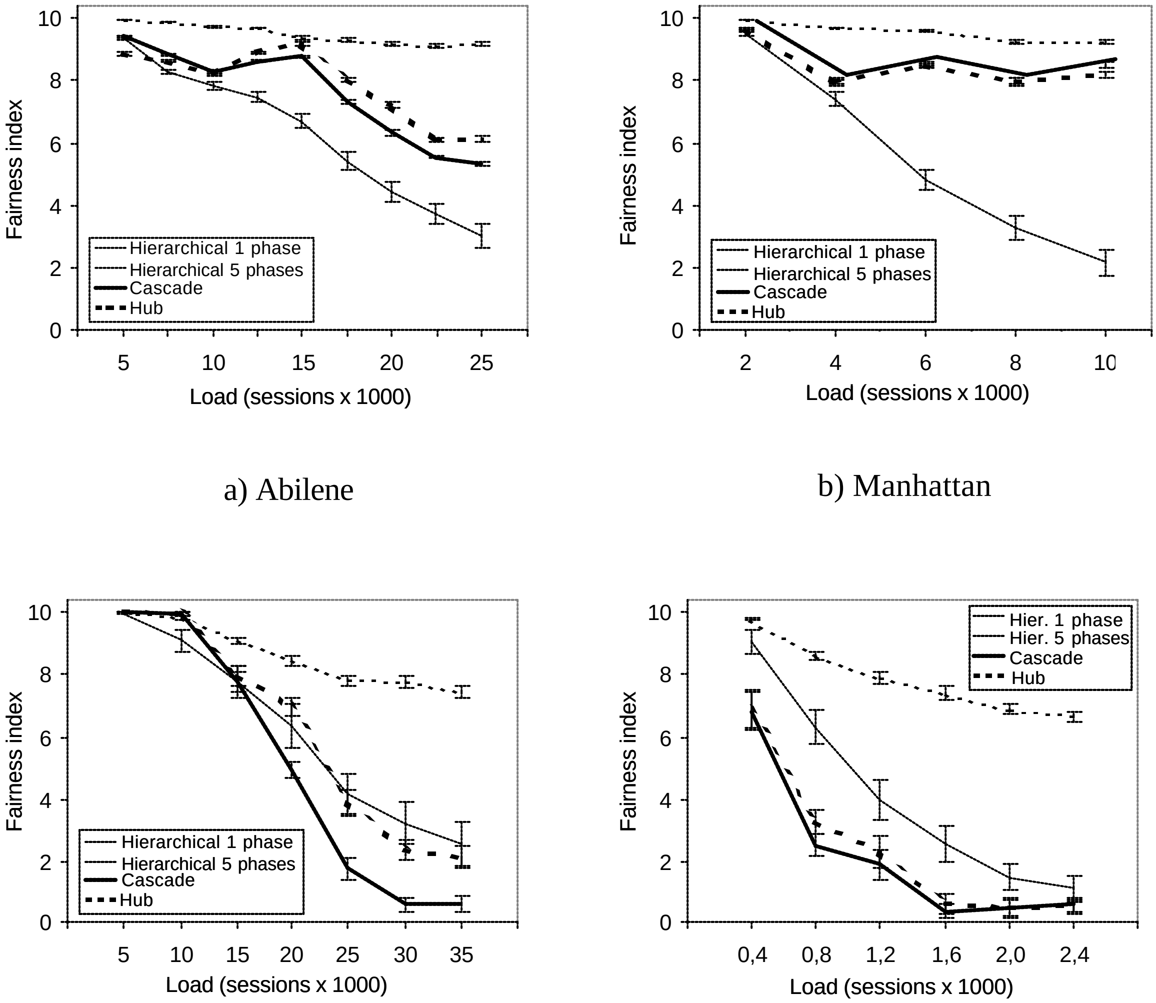
d) RNP2

Figure 6.15 – Comparing negotiation models (voice traffic)

For the hierarchical model, the allocation algorithm works in a centralized way, thus it has a global view of the topology and performs all allocations simultaneously. Domains are chosen in a random order for allocation, so that sometimes the lower capacity domains are tested beforehand, resulting in 100% of granted resources. For the cascade and hub models, there is no such control over the negotiation, which is performed in an asynchronous way.

The next simulations were aimed at comparing results of voice and video services The difference between both services is related to the generated traffic model. In the voice service, calls generate On-Off traffic at 80 Kbps at “On” periods. Video sessions generate VBR traffic with average rate of 384 Kbps for Abilene and GÉANT and 64 Kbps for Manhattan and RNP2. Particular services are important in resource allocation (only throughput, other QoS parameters are not considered) as far as their impact on the traffic prediction that directly affects domain resource requests.

A comparison between pairs of graphs in Figure 6.15 (voice) and Figure 6.16 (video) for each topology shows that index $F_{0.7}$ does not change considerably in order to invalidate conclusions of section 6.4.5. The results obtained for the voice service just come to corroborate with the conclusion that the hierarchical model with 5 phases is able to perform the resource allocation in a fairer fashion than cascade and hub models and the benefit of increasing the number of phases does not compensate the computational cost.



a) Abilene

b) Manhattan

c) GÉANT

d) RNP2

Figure 6.16 - Comparing negotiation models (video traffic)

The conclusion that can be taken by comparing negotiation models in this section is that the hierarchical model with 5 phases produced a significantly higher value of the fairness index than the other models. It was up to 4 times higher in some topologies with high loads. Simulations also showed that the fairness index when used in different topologies is more stable for the hierarchical model.

6.5 Scalability

The Chameleon Architecture is aimed at providing QoS-based services in an Internet-wide scope, although the number of members of a given CDG may be as low as two. Therefore, in order to be able to carry out service negotiations involving small and large CDGs, scalability is a paramount aspect to be considered for a negotiation model. In this section, scalability is evaluated according to three criteria: number of allocation requests, number of messages exchanged and number of peering sessions. For each one, the complexity for the worst case (summarized in) and the values for the four topologies used in the previous sections are presented.

Table 6.4 – Complexity of the scalability

Model	**Criterion**		
	Allocation Requests	**Message Exchanges**	**Peering Sessions**
Hierarchical	$O(n^2)$	$O(n)$	$O(n)$
Cascade	$O(n^2)$	$O(n^3)$	$O(n^2)$
Hub	$O(n^2)$	$O(n^3)$	$O(n^2)$

It is worth to emphasize that in this section (and also in the previous sections) each domain member of a CDG is considered to request services to every other domain.

6.5.1 Allocation Requests

The number of allocation requests generated by a given negotiation model has a huge impact on the scalability of the Chameleon Architecture, since the more requests are sent to be allocated, the more complex and processing intensive becomes the negotiation. As the number of allocation requests grows with the number of domains, it may also interfere with the efficiency and fairness of the negotiation, as shown in sections 6.3 and 6.4.

For the hierarchical model, the main issue is the processing burden imposed by the resource allocation process, since it is carried by centralized entities (the SEs). Let A be the number of requests generated by a given negotiation model. Then, for the hierarchical model $A = \sum_{i=1}^{l} \sum_{j=1}^{S_i} \left(M_{ij} \times \left(M_{ij} - 1 \right) \right) / 2$, where l is the number of levels of SEs (section 5.7.3), S_i is the number of SEs level l, and M_{ij} is the number of participants of SE_{ij}. Participants of a SE are domains for a first level SE and SEs of the immediate lower level for higher level SEs. It is straightforward to see that when there is only one level of SE (that is the case of the evaluations of sections 6.3 and 6.4), that is $l = S_i = 1$, then $A = (n \times (n-1))/2 = C_2^n$, where n is the number of members of the CDG. Hence, there is one request from each domain to every other domain to be processed. This is also the upper bound (i.e., the worst case) for the number of allocation requests in the hierarchical model.

For the cascade and the hub models, the main issue is the cross-interference coming from many domains that harms the efficiency and fairness. In both cases $A = (n \times (n-1))/2 = C_2^n$, that is the worst case also for the hierarchical model. The complexity for C_2^n is $O(n^2)$. For the cascade and the hub models, the number of requests cannot be diminished, whereas in the hierarchical model a SE can be split up in two or more SEs whenever the processing burden is too high.

In the previous sections, the four topologies simulated were configured with only one level of Service Exchange for the hierarchical model. For this reason, the number of allocation requests is the same for the three models, that is $A = (n \times (n-1))/2 = C_2^n$. In this case, the value of A is 66, 120, 325 and 351 for Abilene, Manhattan, GÉANT and RNP, respectively. Scalability aspects related to a 2-level configuration of the hierarchical model are analyzed in section 6.6.4.

6.5.2 Message Exchanges

In order to send requests for service negotiation and receive responses regarding service grants, domains have to exchange protocol messages with each other, in the case of the cascade and hub models and with the SE, when considering the hierarchical model. Each time a message is received by an entity and processed, it is counted as a message exchange, regardless of whether the entity is the final destination. This section considers that the WDS and resource negotiation phases of the negotiation process are being performed together.

For the hierarchical model, each domain sends one request to its SE and receives one response. SEs in turn also send requests and receive answers from their higher level SE. There is no intermediate processing of messages. Let M be the number of protocol messages exchanged and n the number of domains members of a CDG. Then $M = 2n + 2\sum_i^{l-1} S_i$, where

l is the number of levels of SEs and S_i is the number of SEs at level i. When the CDG is configured with just one level of SE, then $M = 2n$. The worst case for the hierarchical model is when each SE has only two participants (one of them can have three participants, in case the number is odd). Then, each SE level contributes to the number of messages with the half of its predecessor level (rounded low), such that $S_i = \lfloor n/2^i \rfloor$. It produces the following series $M = 2n + 2\lfloor n/2 \rfloor + 2\lfloor n/4 \rfloor + \ldots$. Since 3 levels of SEs are expected to be enough for the whole Internet, it does not significantly increase the number of messages. Therefore, the complexity of the number of messages for the hierarchical model is $O(n)$.

For the cascade and hub models, unless each domain has a peering link with every other domain (which is not likely to happen), messages have to be processed by intermediate SBs (Service Brokers) that are in the path between source and destination domains. For the cascade model, $M = 2\sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij}$, where H_{ij} is the number of hops (domains) between domains i and j, if $i \neq j$, and $H_{ij} = 0$ otherwise. The worst case is when the domains are interconnected in a linear way, that is, the outdegree (the number of interdomain links) is 2 for the internal domains and 1 for the two domains in the ends. This type of topology forces the number of messages to be the maximum, because there is always one route for processing each request. Thus, the upper bound is $M = 2\sum_{i=1}^{n} i(i-1)$, because whenever a new domain is connected at one end, there is an increase in the number of messages of $2C_2^n$ since the new domain needs to exchange messages with every other domain, and vice-versa. In this case, the complexity of the number of messages for the cascade model is $O(n^3)$. For the hub model, the number of messages exchanged and the upper bound are twice the number of messages for the cascade model (i.e., the same complexity), since it needs two stages for completing a negotiation.

Table 6.5 depicts the number of messaged exchanged along with the upper bound, for the four topologies simulated in the previous sections and for the three negotiation models. For the hierarchical model, the number of messages is always the upper bound (twice the number of domains), since only one level of SEs was considered. On the other hand, the cascade and hub models exchange much more messages in order to accomplish the same negotiation. It can be observed that the number of messages depends on the number of domains and on the interconnection patterns, more specifically on the network path length (in terms of hops). Thus, for GÉANT the number of messages is higher than for RNP2, even though the former is smaller than the latter (in terms of number of domains).

It is also interesting to observe that the upper bound grows very quickly for the cascade and hub models. The real number of exchanged messages is normally not so high and it decreases proportionally to the upper bound as the number of domains increases. Let P be the ratio between the number of messages and the upper bound. Then, $P = 0.57$ for Abilene, and $P = 0.27$ for RNP2. However, considering that the upper bound for the cascade model is 666,600 messages for 100 domains, and 666,666,000 messages for 1000 domains, even for a small P, the scalability may be a concern for such a model. Therefore, this sample information can be used to take the conclusion that the hierarchical model is also more scalable than the other two models, considering the criterion of the number of message exchanges.

Table 6.5 – Number of Message Exchanges

Topology		**Hierarchical**		**Cascade**		**Hub**	
Name	**Domains**	**Message Exchanges**	**Upper Bound**	**Message Exchanges**	**Upper Bound**	**Message Exchanges**	**Upper Bound**
Abilene	**12**	24	24	656	1144	1312	2288
Manhattan	**16**	32	32	1280	2720	2560	5440
GÉANT	**26**	52	52	3744	11700	7488	23400
RNP2	**27**	54	54	3640	13104	7280	26208

6.5.3 Peering Sessions

In Chameleon, all CDG members must sign a multi-SLA, which automatically exempt them from establishing SLAs with every other member. However, in order to allow messages to be exchanged between negotiating parties, they must be configured to allow the establishment of peering sections. The CDG may opt for a simple policy, whereby negotiations are always accepted, no matter where they come from. As this policy raises serious security implications, domains are likely to prefer to configure manually the peers they accept to exchange negotiation messages with. From the point of view of a single domain, the number of allowed peering sections may vary from 1 to $n-1$, where n is the number of CDG members. Another implication is whether domains must reconfigure their policies for every new CDG member.

For the hierarchical model, each domain must establish a peering section with its SE and SEs with their upper level SE. This represents exactly half of the number of message exchanges (also for the upper bound). Let P be the number of peering sections. Then

$P = M/2 = n + \sum_{i}^{l-1} S_i$, where l is the number of levels of SEs and S_i is the number of SEs at level i. When a new domain becomes member of a CDG, only this domain must configure a new peering session with the SE. The other domains are exempted from doing so.

For the cascade model, only neighbouring domains need to be peers, so that the number of peering sessions is exactly the same of the interdomain links. That is, $P = L$, where L is the number of interdomain links. The upper bound for the cascade model is when the topology is a full mesh, that is, each domain is connected to every other domain[35]. In this case, $P = C_2^n = n(n-1)/2$ and the complexity is $O(n^2)$. A new domain to the CDG implies that only its neighbours have to be reconfigured. The worst case for the cascade model is the normal one for the hub model, as each domain is obliged to become a peer to every other domain. All CDG members have to be reconfigured in the event of new membership.

Table 6.6 shows the number of peering sessions for the topologies simulated in this chapter. As it was shown, the hierarchical model always requires fewer peers to be configured. The cascade model, although its upper bound being high, in practice the number of interdomain links is close to the number of domains. The hub model is the less scalable according to this criteria and it also imposes more configuration work when domains become members of an existing CDG. For 100 domains, P = 4950 and for 1000 domains P = 499,500.

Table 6.6 – Number of Peering Sessions

Topology	Negotiation Model		
	Hierarchical	Cascade	Hub
Abilene	12	15	66
Manhattan	16	24	120
GÉANT	26	37	325
RNP2	27	35	351

[35] This is not likely to happen, though.

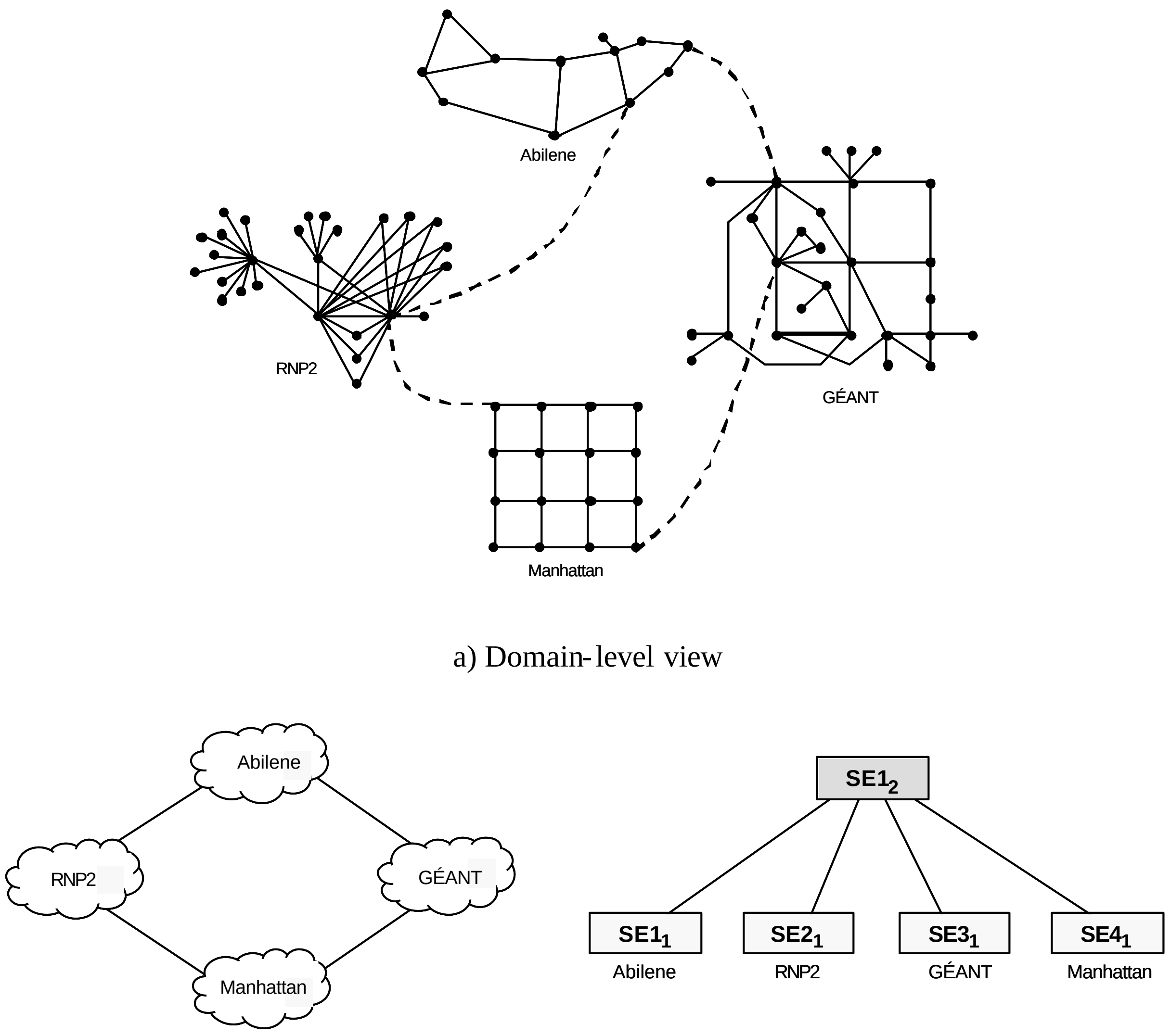
6.6 Higher-Level Hierarchical Negotiation

In the previous sections, the four adopted topologies were evaluated individually, and the hierarchical model was configured with only one level of Service Exchange (SE). In this section, the topologies are unified in a single bigger topology, aimed at evaluating the hierarchical model with two levels of SE_2, according to the description in section 5.7.3. This is mainly because so far the hierarchical model was not fully evaluated. The results of sections 6.3, 6.4 and 6.5 showed that in most scenarios the hierarchical model is able to achieve better efficiency, fairness and scalability than the cascade and hub models.

However, in order to quantify the benefits of scalability, a truly hierarchical scenario have to be evaluated. At the same time, the possibility of such a topology adversely impacting the efficiency of the hierarchical model and the services level experienced by the user must also be considered. One of the goals of this section is taking the first step towards understanding this problem, since there is an obvious trade-off between level of aggregation and fine-grain control over the resource allocation process. Aggregation trades information details for simplicity, so that scalability is achieved but wrong resource allocation decision are therefore more likely to be made.

Figure 6.17a shows the unified topology for this evaluation at the domain level. It is comprised of 81 domains from Abilene, GÉANT, RNP2 and Manhattan, connected in a circular way in order to have two routes through different topologies for any pair of these. For instance, from RNP2 to GÉANT, traffic can traverse either Abilene or Manhattan. Figure 6.17b shows an abstract view of the unified topology, representing the four original topologies as domain clouds. The inter-cloud links are those represented as dashed lines in Figure 6.17a. The two-level SE organization for the hierarchical model is depicted in Figure 6.17c. It consists of four SEs level 1 (SE_1) and one SE level 2 (SE_2).

For this study, only the voice service with exponential call duration time was considered, since the results of sections 6.3 and 6.4 did not reveal significant differences among the services. System load (number of simultaneous calls) was distributed over topologies and domains in a weighted manner, according to their capacity. Abilene, GÉANT, RNP2 and Manhattan were allotted 40%, 50%, 3% and 7% of the total load, respectively. Inside each topology, load was distributed according to the sum of the capacities of the links connected to them. The choice of a destination for a call is handled in such a way that 50% of the calls are headed only to internal domains (within the topology) and the other 50% is distributed among all domains (including the internal ones). In both cases, the probability of a domain being chosen as a destination is relative to its in/out capacity (the same criterion used for distributing system load).



a) Domain-level view

b) Abstract view

c) Two-level hierarchical model

Figure 6.17 – Unified Topology

A very important issue for the hierarchical model is interdomain routing. For a higher level SE organization, it is even more complicated, because the SE_2 sees the SE_1 as domains that have unique routing policies. Since a SE_1 can include several Autonomous Systems (AS) as participant domains, they must agree on a single routing policy before the hierarchical negotiation is started. An adequate solution for this problem is aggregating the participating domains of a SE_1 into a confederation of ASs, as proposed in RFC 3065 [202]. The intention of

BGP confederation is simplifying the complexity of maintaining a large network, which is exactly the same rationale for the scalability of the hierarchical model. As RFC 3065 is a standards track document, it is effectively aimed at being disseminated in the Internet.

6.6.1 Resource Grant Rate

This section presents the evaluation of the negotiation performance, as measured by the resource grant rate (GR), introduced in section 6.3.2. The simulations involved six models: the cascade and hub models; the three versions of the hierarchical model with one level of SE (shortest path, highest throughput and lowest delay algorithms); and the hierarchical model with two levels of SEs (considering only the shortest path algorithm, without optimization). Although the goal is to compare the hierarchical model with one and two levels of SEs, the other models are also presented in order to provide more results.

Figure 6.18 depicts the simulation results, which are similar to those presented in Figure 6.6 and Figure 6.8, whereby the four topologies were evaluated individually. The grant rate curve decreases with the increase in the system load, as expected. The models obtained the same relative results (as in previous evaluations). Buyer domains were granted fewer resources under the cascade and hub models (with the hub model having a slightly better result). The basic hierarchical model (one level without optimization) and its lowest delay optimization also obtained similar results (better than the cascade and hub models). Among the models previously evaluated, the highest throughput optimization of the hierarchical model was able to grant the highest amount of resources. Figure 6.18 does not show the bars for the 99.9 % confidence intervals because they were not large enough to be visualized.

Figure 6.18 also shows that the resource grant rate for the unified topology achieved its best results under the hierarchical model with two levels of SEs. The increase of the 2-level hierarchical model compared to the 1-level one was of 15.3 % for medium system load and up to 22.8 % for a high system load. Compared to the cascade model, the increase was 25.4 % and 50.8 % for the same loads. Although the grant rate is an important index for measuring the success of buyer domains in the negotiation, higher results do not always represent better overall performance. In order to verify if this improvement in the grant rate is also reflected in higher QoS levels, the next sections present the results for the provisioning and admission rates.

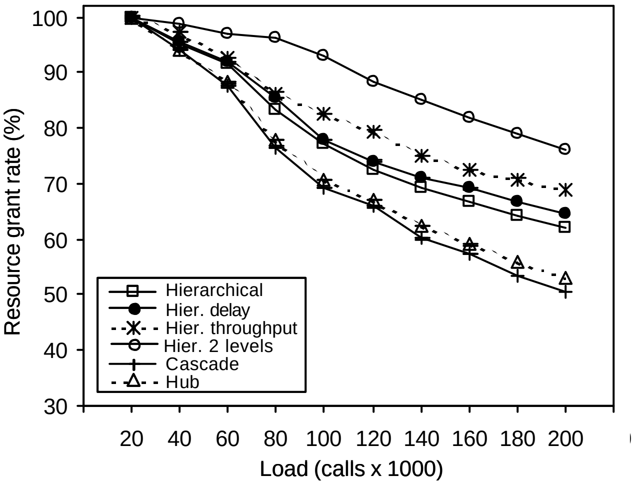


Figure 6.18 – Resource grant rate (GR) - unified topology

6.6.2 Resource Provisioning Rate

The resource provisioning rate (PR), defined in section 0, is another index for evaluating the efficiency in the negotiation, yet from the perspective of service sellers. Figure 6.19 depicts the results of the unified topology for the provisioning rate. Following the grant rate, the results are consistent with those presented in Figure 6.11. In general, the higher the system load, the higher the provisioning rate. In addition, the relative performance achieved by the evaluated models is similar for both grant and provisioning rates.

However, results for grant and provisioning rates differ significantly for the 2-level hierarchical model. For the GR index, the 2-level model obtained the best results, whereas the PR index showed similar results in average for the 1-level and 2-level models. Under 80,000 calls the provisioning rate for the 2-level model stood below the 1-level model and from this load on, they inverted their relative positions. For 20,000 calls, the result for the 2-level model was 10.1 % lower, whereas for 200,000 it was 9.5 % higher. In any case, the 2-level model provisioned much less resources than the highest throughput heuristic of the 1-level hierarchical model. In other words, the 2-level hierarchical models grants more resources to service buyers, but it uses less resources of service sellers, causing a lower provisioning rate.

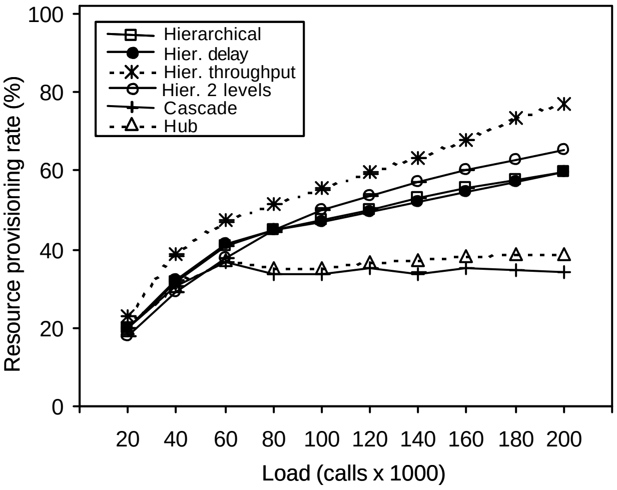


Figure 6.19 – Resource provisioning rate (PR) unified topology

The explanation for this apparently weird behaviour stands in the method the level 1 SEs use for resource distribution after the higher-level SE_2 finishes the resource allocation process. This procedure is the step 7 of the hierarchical negotiation round, described in section 5.7.3. In the evaluated scenario, each domain and interdomain link receive the same percentage of resource that was provisioned to the abstract topology with higher-level aggregate information. For instance, let us suppose that the SE_2 allocated 50% of the resources offered for $SE1_1$ (Abilene). Consequently, 50% of the available resources[36] will be provisioned by the $SE1_1$ in each participant domain and interdomain link. Since this is a very simple and not optimized method for distributing resources, it generated a lower level of resource provisioning[37].

6.6.3 Connection Admission Rate

Grant rate and provisioning rate are two important indexes for quantifying the success in the negotiation of both service buyers and sellers. For this reason, they were presented as goals of the hierarchical negotiation in section 5.7 and so far, they were used to measure efficiency. However, as the 2-level hierarchical model uses information aggregation for obtaining scalability, it becomes essential to check whether this model is able to perform the correct resource provisioning. In other words, this section is aimed at verifying whether the lack of fine-grain information during the negotiation will not cause an unexpected violation of the QoS guarantees of the end-to-end services.

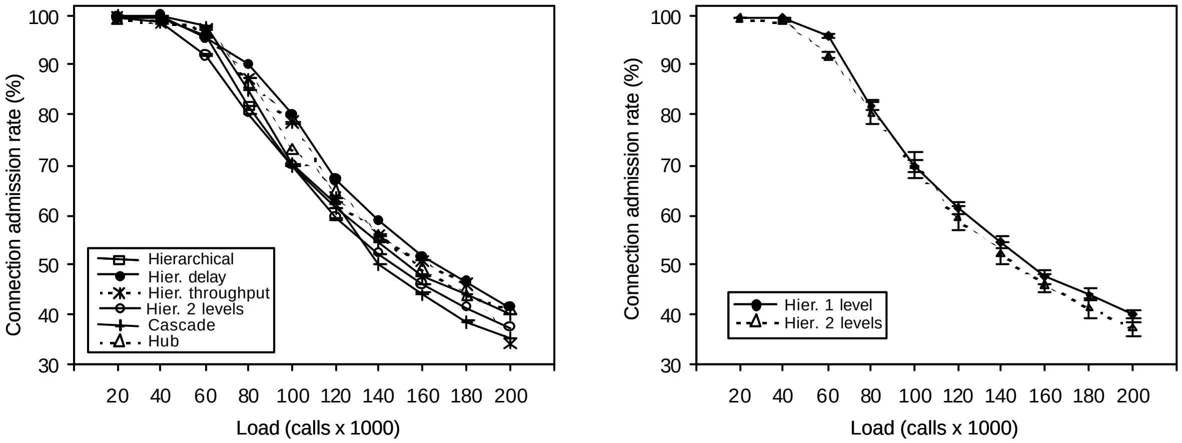
[36] Resources are also provisioned during the local negotiations (step 3 in section 5.7.3)

[37] Provisioning strategies for the hierarchical model are an important area of further work.

The best approach to do this would be by measuring the QoS performance parameters, throughput and delay, of the voice traffic defined for the sample WDS used in this study. This is not possible, though, since no packets are actually generated. Consequently, a new index was created for measuring service performance, which is the end-to-end connection admission rate. Whenever a new voice call is generated, the CAC mechanism implemented by the Service Broker checks for resource availability in every domain along the path from source to destination [129]. If there are enough resources, the available resource count is decreased and the call is admitted. Otherwise, it is blocked. When a call is finished, a teardown message is sent by the CAC mechanism and the available resource count is increased. At all times, the simulator maintains information on the number of generated and admitted calls. This end-to-end CAC mechanism is used simply for evaluation purposes. It does not make sense to implement it in a real environment, since the volume of signaling messages would be enormous, comparable to IntServ (although RSVP messages are processed by every router in the path). A real implementation is likely to be based upon a local CAC mechanism.

Let C_i and A_i be, respectively, the number of calls generated in i $(i = 1,2,\dots,n)$ and admitted by the end-to-end CAC mechanism. The global connection admission rate is defined as $AR = 100 \times 1/n \sum_{i=1}^{n} A_i / C_i$. The results of this section refer to the average of AR for the 100 replications of the simulation.

Figure 6.20a depicts the simulation results of the connection admission rate for the unified topology including the six models evaluated in this section. It can be observed that the curves are very close to each other, unlike the graphs for the grant and provisioning rates. The best results for the admission rate were obtained by the lowest delay optimization of the 1-level hierarchical model. This is a very instigating point, since the efficiency of this optimization measured by both GR and PR indexes did not reveal encouraging results in the previous sections. The hub model also achieved positive results (better than those of the 1-level and 2-level hierarchical models). Furthermore, additional simulations were undertaken considering a varied number of phases (section 6.4) in the resource allocation of both 1-level and 2-level models. The results revealed an increase in the performance of the admission rate, showing another benefit of dividing the allocation process into phases, in addition to its intended use for obtaining fairness in the negotiation.



a) b)

Figure 6.20 – Connection admission rate – unified topology

The most important result of the admission rate for validating the 2-level hierarchical model is presented in Figure 6.20b. The graph shows only the curves for the 1-level and 2-level configurations of the hierarchical model, with the vertical bars representing the 99.9% asymptotic confidence intervals. With high load, the 1-level model was able to admit up to 6.5% more calls than the 2-level model. However, for medium load, both obtained very close results. In addition, the graph shows that the bars are overlapped for almost all loads, hence giving the statistical guarantee of the two models being nearly indistinguishable, as far as the connection admission rate is concerned. Since here this index is measuring the service quality level provided by the network to its users, a possible conclusion of this study is that the 2-level model is a viable configuration, since it is expected to provide increased scalability while at the same time maintaining the agreed QoS guarantees.

6.6.4 Scalability Issues

Scalability is the sole reason for justifying the additional complexity of implementing a higher-level configuration for the hierarchical model. Otherwise, a single level centralized model will be the best option, as the results of sections 6.3, 6.4 and 6.5 attest. Therefore, these numerical results are presented for the unified topology concerning number of allocation request, message exchanges and peering sessions.

Table 6.7 presents scalability results for the 1-level and 2-level configuration of the hierarchical model and the cascade and hub models. It corroborates the results of section 6.5, showing that the hierarchical model (both 1-level and 2-level) has a huge advantage over the cascade and hub models in terms of scalability, in particular with respect to the number of messages that must be exchanged by protocol entities.

Table 6.7 – Scalability for the unified topology

Model	Allocation Requests	Message Exchanges	Peering Sessions
Hierarchical 1-level	3240	162	81
Hierarchical 2-level	862	170	85
Cascade	3240	67.828	115
Hub	3240	135.656	3240

Comparing the two configurations of the hierarchical model, it can be observed that the 2-level model requires slightly more message exchanges and peering sessions, due to the interconnection of SEs_1 with the SE_2. However, this model generates 74.4% less allocation requests than the 1-level one, thus drastically reducing the burden on the Service Exchanges for processing the same service requests. Finally, a conclusion would be that the benefits of using the hierarchical model are more evident for large CDGs, which effectively implement the higher-level SEs required by the hierarchical structure.

6.7 Other Criteria for Comparison

In the previous sections, the cascade, hub and hierarchical models were compared according the criteria of efficiency, fairness and scalability. The simulations and analyses have shown that the hierarchical model outperforms the other in the scenarios considered in this study. However, other aspects have to be taken into account when choosing a negotiation model for deploying the Chameleon architecture that can point out different conclusions. This section presents other criteria and comments briefly on their influence on the three negotiation models.

6.7.1 Reliability and Resilience

The reliability of a negotiation model is the extent to which a server outage interferes in the service negotiation of the whole CDG. Centralized solutions generally raise concerns with respect to reliability, and special attention must be given in order for the service to be available whenever it is needed. Therefore, in a simple analysis the distributed negotiation style could be considered more reliable than the centralized one. The main drawback is that in a centralized model there is a single point of failure, such as the SE in the hierarchical model. If the SE fails, services cannot be further negotiated until it is running again. A hierarchy of SEs can improve the situation in a certain way, since if a SE fails, it does not affect the operation of the other ones. The failure of an upper level SE blocks inter-area negotiations but does not interfere with intra-area negotiations.

Anyhow, this type of fragility is not desirable, since a part of the Internet could stop deploying advanced services for a while when the SE is down. However, this situation is neither unusual nor the first in the Internet. The DNS (Domain Name Service) has a hierarchical structure and it works well by means of server replication. Route Servers also have replicated servers for enhancing their reliability [97][144]. Hence, the implementation of a SE can also be based on replicated servers, which can be located in different geographical places, similarly to the DNS. This solution imposes additional complexity and costs but it is worth adopting because it makes the hierarchical model more reliable.

The cascade and hub models are based on distributed negotiations, thus not having the same problems of the hierarchical model. The failure of a SB in the cascade model or the Service Provider in the hub model can cause some problems, but clearly over a much more limited scope. Since they are by nature more reliable than the hierarchical model and do not need complex solutions for it, some small CDGs may opt for using them, even considering the advantages of the hierarchical model in relation to efficiency, fairness and scalability.

Resilience refers to how fast a negotiation model recovers from link outages or sudden route changes. Once the negotiation is completed and resources are allocated, the services should be available during the time period of the service schedule (section 4.3.1) or until a new request is sent by the buyer domain. If any problem that could interfere in the QoS guarantees provided to the service happens in the meanwhile[38], the negotiation model must be able to reestablish the normal service operation as quickly as possible. This ability is strictly related to the way they deal with urgent negotiations. The cascade and hub models do not have restrictions for urgent negotiations. Therefore, in the case of a route change, as soon as the new route is computed, the service can be renegotiated and the service can be reestablished, as long as there is at least one end-to-end wherein the service request can be accommodated. In the hierarchical

[38] Domains become aware of problems in the end-to-end resource allocation by means of the metric interchange function of the monitoring plane, which certainly will report violations of QoS targets.

model, recovering from a service outage may take more time, since the SE must first be notified of the new topology. Then it has to find out all requests that were affected by the outage and try to reallocate them accordingly.

6.7.2 Financial Incentives

In the current Internet, the interconnection of domains faces some problems, as shown in section 2.1.7, notably the lack of financial incentives for upgrading interdomain links that could otherwise ameliorate at a great extent the global performance of the Internet. Independently of the technical benefits that a negotiation model provides, if it is not interesting from a commercial point of view or if it does not help in resolving old known problems, it will not be adopted by any CDG. So far in this chapter, it was assumed that domains were willing to deploy QoS-based services by means of the Chameleon architecture. Therefore, the technical evaluations presented some guidelines for helping them in choosing the more suitable negotiation model.

When domains are not sure of deploying advanced services and using the Chameleon Architecture, a negotiation model that is able to provide the adequate financial incentives may be a determining factor for convincing them. From this point of view, the cascade and hub models do not do a good job, as they rely on existing interconnection structures and routing policies.

On the other hand, the choice of the hierarchical model may be a factor of success, by giving the right incentives, even for resolving the current problems of domain interconnection. Firstly, in the hierarchical model there is no distinction between transit and peering links. Once an interdomain link is informed to the SE as belonging to the CDG topology, it can be used for selling and buying services. Regardless of how large a domain is, when it buys a service it is seen as a client. Similarly, even small domains behave as transit domains when they sell services. As domains are remunerated by all the traffic they have sold, they have interest in keeping the performance guarantees within the bounds specified in the WDS. The common practices of “hiding” some peering links from other peers or not upgrading congested peering links make no sense in this scenario anymore. Even multi-homed domains that normally do not forward transit traffic could be benefited, as they can sell the idle capacity of their interdomain links.

Secondly, in the hierarchical model there is no difference between the exchange-based or the direct-circuit interconnection models. Currently, Internet exchanges are always congested, because they do not have incentives for upgrading their network facilities. In the hierarchical model, they can be seen as another seller domain that interconnects various domains and provides various internal paths for deploying WDSs. According to this modeling, exchanges do not charge by the physical capacity of the links, but by the quantity of traffic they sell.

A more detailed investigation of financial incentives for dynamic service negotiation models is out of the scope of this thesis, and therefore is left as a future work (section 7.2).

6.7.3 Complexity and Costs

One of the main criticisms against the introduction of QoS in the Internet is that it is not worth the additional complexity required for implementing it. According to this opinion, the Internet is robust and scalable because it is simple, and nothing should be done for changing it. The same reasoning can be extended to negotiation models. In section 5.2, it was shown that dynamic service negotiation is necessary, although it is more complex than static negotiation. Therefore, the challenge is to find a negotiation model that does not add too much complexity and costs.

The hierarchical model is more complex than the cascade and hub models, because it creates a structure of SEs that otherwise would not be necessary. Nevertheless, it was shown in section 6.5 that it is more scalable than the other two models. The installation and maintenance of the SEs adds some extra costs to the domains, in case the CDG is truly collaborative and the domains share the its costs of deployment. On the other hand, they save money, since the hierarchical model is more efficient in discovering and utilizing existing resources. In a competitive environment, some CDGs may be truly business-oriented companies that will assume the costs of operation it they envision a reasonable revenue in return. In this case, cost is not a problem for the hierarchical model, as long as the advantages compensate for it.

6.8 Summary

In this section, some findings of the comparison of negotiation models undertaken in this chapter are reported. The soundest conclusions came from the evaluation of the criteria of efficiency, fairness and scalability. The other criteria can only be used as additional information.

- The cascade and the hub models achieved very similar results in the simulation study, for both efficiency and fairness. Usually the hub models had slightly better results, mainly because it takes more time for doing bad allocations (due to its two negotiation phases). Since the cascade model is more scalable than the hub model, the former is preferred whenever a distributed model is to be used.

- The hierarchical model is by far the best alternative, when comparing efficiency, fairness and scalability, even for the routing passive style. It is more efficient than the other two models due to the centralized negotiation process. The negotiation algorithm has a global view of the topology and performs all allocations simultaneously, thus making precise allocations. Both service buyers and sellers obtain better deals from the negotiation, as the results for the grant and provisioning rates show in section 6.3. It also achieves higher fairness, when at least five phases are used by the allocation algorithm. The reason is that the difference in resource distribution among domains is considerably smoothed as the number of phases is increased. As far as scalability is concerned, the advantages are obvious, coming from the very nature of the hierarchical organization of the Service Exchanges, which requires a smaller number of messages to be exchanged. In addition, this chapter showed that it is better suited to providing the right financial incentives for deploying QoS and resolving some of the current problems of the interconnection of domains in the Internet. On the other hand, the hierarchical model is more complex due to the implementation and maintenance of the SE structure as well as being less reliable with its SE as a single point of failure.

- The comparison of the two optimizations for the hierarchical model showed that lowest delay heuristic did no bring any advantage over the basic shortest-path heuristic. On the other hand, the highest throughput heuristic was able to obtain reasonable gains in efficiency. Whether the additional complexity of adopting the routing active style is worth the benefits of the highest throughput heuristic, is up to each CDG. A topic of further study is investigating most efficient optimizations, such as those described in section 5.7.4.

- One simple conclusion is that the cascade model is the most adequate for small CDGs, due to its simplicity and the hierarchical model for large CDGs.

Chapter 7

Conclusions and Future Work

Services based on QoS guarantees have increasingly been attracting the attention of the Internet community for the last years. Currently, the scenario seems to be favorable for QoS deployment: we have users willing to utilize multimedia applications with an adequate quality, IETF standards for providing QoS, commercial products that implement those standards along with other proprietary solutions and providers that are interested in improving their profits by offering value-added services. There is also a trend for evolving the Internet into a true converged network, capable of combining traditional data applications with voice and video applications. Nonetheless, QoS is not yet a reality in the Internet. There is no well-developed solution for the interconnection of domains, so that they have the right financial incentives for providing performance guarantees based on QoS metrics for some part of the traffic that traverses their networks.

Although the simplicity of the IP protocol is the main reason for the scalability and robustness of the Internet, some extensions to the Internet infrastructure are needed in order to make it possible to resolve both challenges of deploying advanced services and giving the right incentives to users and domains. Sometimes over-provisioning is presented as the solution, but this approach has not been successful in solving the afore-mentioned problems. At most, over-provisioning provides a means for large domains having a network nearly free of packet loss.

In the sequence of this chapter, section 7.1 summarizes the contributions of this thesis. Section 7.2 present directions for future work. Finally, section 7.3 presents the final thoughts and conclusions.

7.1 Summary of Contributions

This thesis contributes with some step towards a scenario where the Internet is able to deploy end-to-end advanced services based on QoS guarantees. The major contributions of this work can be grouped into five topics. This research also produced two minor contributions.

7.1.1 The Design of the Chameleon Architecture

The high level design of the Chameleon architecture was presented in Chapter 3. Chameleon gives a step forward of the current proposal for Internet QoS by clearly separating the functions particular to abstract services to their implementation with QoS technologies. The service and operation planes are logically isolated from each other, and the communication between them happens through well-defined interfaces, following the usual approach of isolating layers in network architectures. This separation makes it possible to a domain to choose any approach for obtaining QoS, as long as it is able to provide services with the required performance guarantees. The deployment of transport services is completed with the monitoring plane, which collects information from the operation plane, makes some analysis and gives relevant feedback results to the operation and service planes for on-the-fly service tuning.

The concept of Chameleon Domain Group (CDG) represents also a contribution associated with the design of the Chameleon architecture. It was used all over this thesis because it showed useful for referring to a group of domains that deploys de Chameleon architecture.

7.1.2 Service Definition based on Well-Defined Services

Chapter 4 presented Chameleon's approach for service definition, which is based on Well-Defined Services (WDS). A WDS combines some very useful features that are only partially covered (or not at all) by other approaches found in the literature: a) A WDS gives a precise definition of service semantics and it is not tied to a particular QoS technology. Packets can be easily identified as belonging to a particular WDS and given the right treatment in order to preserve the end-to-end service semantics; b) WDSs provide flexibility in creating service classes and instances. Although domains do not have the freedom for negotiating ad-hoc services (as in Tequila's SLS), they can create WDS instances with different parameters; c) It is aimed at facilitating service negotiation by a high number of domains simultaneously; d) It supports negotiation of a service as a whole, including bidirectional traffic.

7.1.3 A Structure for Service Negotiation

In Chapter 5, some concepts related to service negotiation in Chameleon were structured in a logical way. The work on dynamic interdomain service negotiation for the Internet is very recent. Most of the existing proposals focus on the negotiation between a provider and its client corporate networks, because it is a short-term necessity of the ISPs for automating their service offerings. The work presented in Chameleon can positively contribute for the development of solutions for the negotiation among domains the Internet backbone.

The most distinctive features of the Chameleon approach for service negotiation are: a) clear separation between user and transport service negotiations; b) the negotiation process, comprised of SLA, WDS and resource negotiation phases; c) a simple taxonomy for classifying negotiation models based on negotiation styles; d) adaptation of negotiation models used in other contexts to the Chameleon architecture, namely, the cascade, hub, wave and border models; e) interaction among different CDGs, which can use homogeneous or heterogeneous negotiation models.

7.1.4 The Proposal of the Hierarchical Model

The hierarchical model was described in section 5.7, together with the other four models that were adapted to the Chameleon's approach for service negotiation. However, it was given special attention, as it is considered the most suitable for service negotiation in the Chameleon architecture. The hierarchical model is centralized and it introduces the new concept of Service Exchange (SE), which is an entity that coordinates the service negotiation process among domains of a CDG. The SE has enough information about the topology and the purchase and sale interests. Therefore, it is able to perform near optimal choices and use optimized algorithms for resource allocation. SEs aggregate requests when the destination is outside the SE area and send them to a higher-level SE, due to their hierarchical structure. This model has therefore the potential scalability for dealing with service negotiation in the whole Internet, as the DNS system does for the current Internet.

7.1.5 A Comparative Evaluation of Negotiation Models

An evaluation of the cascade, hub and hierarchical negotiation models was carried out and the results were reported in Chapter 6. Six criteria were used for comparison: efficiency, fairness, scalability, reliability and resilience, financial incentives, and complexity and costs. The efficiency and fairness criteria were evaluated by a simulation study. Two indexes were developed for the efficiency criterion: resource grant rate and resource provisioning rate. For the fairness criterion, a fairness index was proposed, as well as solution for obtaining fairness for the hierarchical model was made. The scalability criterion considered three sub-criteria: number of resource allocations, number of message exchanges and number of peering sessions. The three negotiation models were compared according to a simple analytical modeling and to the simulation results. The remaining three criteria were used only in a textual, argument-based comparison. Four topologies were used in the comparisons, namely, Abilene, GÉANT, RNP2 and Manhattan. Some findings of the comparisons are:

- The cascade and hub models achieved very similar results in the simulation study, for both efficiency and fairness. Usually the hub model had slightly better results, mainly because it takes more time for doing bad allocations (due to its two negotiation phases). Since the cascade model is more scalable than the hub model, the former is preferred whenever a distributed model is to be used.
- The hierarchical model is by far the best alternative, when comparing efficiency, fairness and scalability, even for the routing passive style. It is also able to provide the right financial incentives for deploying QoS and to resolve some of the current problems of the interconnection of domains in the Internet. On the other hand, the hierarchical model is more complex due to the implementation and maintenance of the SE structure and also less reliable because the SE is a single point of failure. As far as efficiency is concerned, the increase in the grant rate produced by the hierarchical model compared to the cascade model was from 15% to 50% under high load and from 3.3% to 15.8% under medium load. The increase in the provisioning rate was from 20% to 145% under high load, and from 9% to 33% under medium load. For the fairness, the hierarchical model with 5 phases was always classified as desirable and acceptable, even in situations where the other models fell down to unacceptable levels.
- The comparison of the two optimizations for the hierarchical model showed that lowest delay heuristic did not bring any advantage over the basic shortest-path

heuristic. On the other hand, the highest throughput heuristic was able to obtain reasonable gains in efficiency. It obtained increases over 10% (up to 14.9%) with high load in the resource grant and up to 30% in the resource provisioning. Whether the additional complexity of adopting the routing active style is worth the benefits of the highest throughput heuristic, this is left up to each CDG to decide. A topic of further study is investigating more efficient optimizations, such as those described in section 5.7.4.

One simple conclusion can be seen as that the cascade model is the most adequate for small CDGs, due to its simplicity and the hierarchical model for large CDGs.

This simulation study also helps to understand some aspects related to the performance of the four adopted topologies. In this sense, it represents a specific contribution to the Brazilian research and academic network, namely the RNP2.

7.1.6 Minor Contributions

Two minor contributions of the development of the Chameleon architecture were presented in Chapter 2, as a background for the rest of the document.

- The approach for dealing with the QoS problem was by focusing on the characterization of services and the current problems of the interconnection of domains in the Internet. This approach explicitly recognizes that there is currently little or no support for advanced services in the Internet, and that the current interconnection structure does not provide the right incentives for deploying QoS. On the other hand, most works on QoS focus on the technologies as the main background [45][72][160][200][220].
- The service life cycle model, presented in section 2.2.6, which is comprised of service definition, implementation, negotiation, provisioning, utilization, creation, monitoring and accounting. This life cycle oriented the design of the Chameleon architecture and had a fundamental importance in focusing this thesis in service definition and negotiation. A comprehensive understanding of the life cycle of advanced services in the Internet, having in mind that any successful solution has to be driven by sound business models and not in the enabling technologies, is paramount for the deployment of QoS solutions for the whole Internet.

7.2 Future Work

The work so far in the Chameleon architecture has covered important issues on the deployment of end-to-end QoS. However, since Chameleon defines a complete architecture encompassing most of the phases in the service life cycle (according to our proposal in section 2.2.6), there are some important points that deserve a more in-depth proposal and evaluation, but unfortunately were not sufficiently elaborated because they fell out of the scope of this thesis. On the other hand, some aspects covered in Chameleon open the path to many research areas. This section presents some directions for future work identified during the development of the Chameleon architecture.

1. Support for the Hose and Funnel scope types: Scope types in Chameleon are related to the number of source and destination domains that are enumerated in the SLS field "scope". As stated in section 1.3, only the Pipe scope type is considered in this thesis. In the Hose type, which has one source to multiple destinations, resource utilization may be distributed in any way to destination domains. In other words, the amount of resources that must be provisioned is not known beforehand. There is no predefined amount of traffic to be sent to every other destination domain. The Funnel type is the opposite, with many sources for one destination. The Chameleon architecture must be adapted to accept the Hose and Funnel types, since negotiations are based on traffic predictions in a one-to-one basis. It is not simple to perform true end-to-end negotiations for scope types where there is no precise amount of resources to be negotiated between every pair of domains. It affects both the service negotiation process and the resource provisioning strategy.

2. Definition of APIs for the internal interfaces: In section 3.1.4 the S-O, S-M and O-M interfaces were presented. Further, in sections 3.2 (service plane), 3.3 (operation plane), and 3.4 (monitoring plane), the relationships between the planes were also elaborated. However, the APIs for those interfaces were not formalized. The further developments of the implementation of the Chameleon architecture would be benefited, in case the APIs were made available.

3. Design of protocols for the negotiation models: the messages needed for implementing protocols for the cascade, hub, hierarchical, wave and border models were described in sections 5.5.2, 5.6.2, 5.7.5, 5.8.2 and 5.9.2, respectively. The deployment of service negotiation in the Internet requires well-specified protocols,

though. An important future work is to design protocols for each model (mainly for the hierarchical model), including a review of the messages, the format of each message according to their TLVs (Type, Length and Value), rules for the message processing, state machines for all types of communicating entities. This information is generally required in the specification of a protocol.

4. Extending the evaluation for the wave and border models: The comparative evaluation of Chapter 6 considered only the cascade, hub and hierarchical models. The main reason was that the wave and border models have slightly different characteristics, thus they were not included in the evaluation due to scope (and time) constraints.

5. Implementing more efficient methods for resource allocation: The hierarchical model permits the utilization of practically any known heuristic or deterministic method for resource allocation in computer networks. The problem of allocating resources among domains can be mapped to the most common problem of allocating resources inside a network, wherein domains are considered routers. In section 5.7.4, some alternatives are described, such as the Constraint Satisfaction Problem based on backtracking or the Blocking Island paradigm. However, in Chapter 6 only two variations of the basic shortest-path heuristic based on QoS Routing optimizations were evaluated. Even though they presented significant improvements over the basic model, more efficient heuristics can emphasize the benefits of the hierarchical model.

6. Evaluating the resource provisioning according to the fairness criterion: The fairness index proposed in section 6.4.3 and used for the evaluation of fairness considers the resource grant, i.e., it measures the level of fairness related to resource distribution to buyer domains. An interesting result would be obtained by comparing it with the fairness for seller domains. In other words, adapt the fairness index for resource provisioning.

7. Using the max-min fairness criterion for service negotiation: Fairness in bandwidth allocation traditionally has been evaluated by the max-min fairness criterion [117], which determines that an allocation is fair when it maximizes the resources (max) assigned to those users who receive the lowest quantities (min). A max-min fairness index for service negotiation could be defined with the following steps: a) build a vector of feasible allocations for each mode; b) classify the vectors in ascending order, according to the max-min allocation; c) assign a number tor each model, according to the order; d) normalize this number by the number of models considered. This

procedure creates a value between 0 and 1 for each model, which is the max-min index. The difference between this approach and the fairness index proposed in section 6.4.3 is that the former uses directly the amount of granted resources (in bps), whereas the latter considers the granted percentages. A key benefit of the approach adopted in this thesis is isolating fairness analysis from efficiency in resource allocation. Additionally, is has other advantages: (1) it is easier to calculate, that is, it has a higher computational efficiency; (2) it can be calculated for each negotiation model or allocation algorithm separately from others; (3) it generates absolute values, contrasted to relative comparisons of the max-min criterion. In any case, comparing our fairness index with a max-min index could yield often new insightful results.

8. Comparing negotiation models through economic models: In section 6.7.2, some arguments showing that the hierarchical model is able to provide better financial incentives that the other models were presented. However, conclusive evidence would need a more in-depth evaluation of these aspects. Service pricing, as described in section 2.2.6, and economic models and tools should be used [21][62] (such as game theoretic analyses and auction-based approaches [178]).

9. Formal specification of the Chameleon architecture: The design of the Chameleon architecture presented in this document was based on textual description along with elucidative pictures. However, no attempt was made towards making a formal specification of Chameleon's properties, through a suitable language, such as LOTOS [58]. The formalization is useful for making it possible to verify important aspects of the architecture, such as the interfaces (section 3.1.4), service definition (Chapter 4) and service negotiation (Chapter 5). Furthermore, the mapping process from the abstract functions of the service plane into practical implementations of the operation plane will be greatly benefited with the aid of a formal specification.

7.3 Final Conclusions

This research started as a simple architecture for providing a flexible compromise between the rigid per-flow guarantees of IntServ with the looser aggregate-based guarantees of DiffServ. Although it seemed to be promising, soon it was recognized that deploying advanced services is not a matter that can be solved only by implementing mechanisms for resource reservation or service differentiation at the router level. Therefore, a solution at a higher level of abstraction was sought, and the Chameleon architecture was designed as an overlay network.

Regarding the deployment of the Chameleon architecture, some final thoughts are worth to be mentioned. The main strength of Chameleon is giving special attention to the abstract aspects of services (i.e., service definition and negotiation), and placing them in a separate plane. Should the Chameleon architecture be deployed in the Internet, it is expected to happen gradually. Some different evolution scenarios can be conceived, as far as the availability of advanced services based on WDSs and dynamic service negotiation are concerned.

The first scenario is the current Internet, with no advanced services and no dynamic negotiation. It has known advantages and limitations. The second scenario assumes that only advanced services are available. There is an obvious advantage of providing QoS guarantees. However, it is likely to be difficult to deal with, since it cannot make use of the flexibility that the dynamic negotiation provides. In the third scenario, dynamic negotiation with the hierarchical model is available for the best effort service. This scenario is not the ideal, since it does not presuppose advanced services, but it can help domains in resolving their current interconnection problems. In the fourth scenario, both advanced services and the hierarchical model are deployed. This scenario is expected to produce a synergetic effect for leveraging the Chameleon architecture. The fifth scenario is an upgrade of the fourth scenario for the routing active style, which allows the utilization of more efficient heuristics for resource allocations. It was shown in our simulations, that there are advantages in this scenario. Other scenarios are possible, involving different negotiation models. It is paramount that the scenarios are well designed in a way that the deployment of the Chameleon architecture is not blocked by psychological barriers or lack of financial incentives.

The work presented in this document establishes a sound foundation for further studies on interdomain service definition and negotiation and their relationship with intradomain activities required for service implementation and provisioning. It also calls the attention to the fact that services are at the heart of the Internet QoS problem, not technologies.

References

[1] Abarbanel, B. & Venkatachalam, S., BGP-4 support for Traffic Engineering, Internet Draft, <draft-abarbanel-idr-bgp4-te-02.txt>, work in progress, December 2000.

[2] Abilene NOC, Abilene Connection Traffic Statistics, http://stryper.uits.iu.edu/abilene, last visited at 21/01/2003.

[3] Akamai Technologies, Internet Bottlenecks: the Case for Edge Delivery Services, White Paper, 2000.

[4] Allard. H., Understanding QoS. How to obtain quality over bursty networks, *Teleconference Magazine*, **9** (1), January 2000.

[5] Allen, D., The Impact of Peering on ISP Performance: What's the Best for You?, *Network Magazine*, http://www.networkmagazine.com/article/NMG20011102S0006, November 2001.

[6] Allen, D., Qwest Introduces On- and Off-Net SLAs, *Network Magazine*, http://www.networkmagazine.com/article/NMG20020104S0006, January 2002.

[7] Almes, G., Kalidindi, S. & Zekauskas, M., A One-way Delay Metric for IPPM, RFC 2679, September 1999.

[8] Almes, G., Kalidindi, S. & Zekauskas, M., A One-way Packet Loss Metric for IPPM, RFC 2680, September 1999.

[9] Almes, G., Kalidindi, S. & Zekauskas, M., A Round-trip Delay Metric for IPPM, RFC 2681, September 1999.

[10] Andersen, D., Balakrishnan, H., Kaashoek, F. & Morris, R., Resilient Overlay Networks, in Proceeding of the 18th ACM Symposium on Operating Systems Principles, Banff/Canada, October 2001.

[11] Anderson, L., et al., LDP Specification, RFC 3036, January 2001.

[12] Angel, J., Toll Lanes on the Information Superhighway, *Network Magazine*, http://www.networkmagazine.com/article/NMG20000517S0170, February 2000.

[13] Apostolopoulos, G. Guerin. R., Kamat, S., & Tripathi, S.K., Quality of Service Based Routing: A Performance Perspective, in Proceedings of the ACM SIGCOMM'98, Vancouver, Canada, pp. 17-28, September 1998.

[14] Apostolopoulos, G., QoS Routing Mechanisms and OSPF Extensions, RFC 2676, August 1999.

[15] AQUILA Project, http://www-st.inf.tu-dresden.de/aquila, last visited at 21/01/2003.

[16] Asgari, A. et al., A Monitoring and Measurement Architecture for Traffic Engineered IP Networks, in Proceedings of the First International Symposium on Telecommunications (IST2001), Tehran/Iran, September 2001.

[17] Aurrecoechea, C., Campbell, A T. & Hauw, L., A Survey of QoS Architectures, *ACM Multimedia Systems Journal: Special Issue on QoS Architectures*, **6**(3), pp. 138-151, May 1998.

[18] Awduche, D. et al., Requirements for Traffic Engineering over MPLS, RFC .2702, September 1999.

[19] Awduche, D., MPLS and Traffic Engineering in IP Networks, *IEEE Communications Magazine*, **37** (12), pp. 42-47, December 1999.

[20] Awduche, D. et al., Overview and Principles of Internet Traffic Engineering, RFC 3272, May 2002.

[21] Baake, P. & Wichmann, T., On the economics of Internet peering, *Netnomics*, **1**, pp. 89-105, January 1998.

[22] Bajaj, S., Breslau, L. & Shenker, S., Uniform versus Priority Dropping for Layered Video, in Proceedings of the ACM SIGCOMM'98, Vancouver, Canada, pp. 131-143, September 1998.

[23] Baker, F., et al., Requirements for IP Version 4 Routers, RFC 1812, June 1995.

[24] Baker, F., The Case for QoS, *Cisco Systems Packet Magazine*, pp. 62-67, 4th Quarter 2000.

[25] Bernet, Y. et al., A Framework for Differentiated Services, Internet Draft, <draft-ietf-diffserv-framework-02.txt>, work in progress, February 1999.

[26] Bernet, Y., Smith, A., Davie, B., Specification of the Null Service Type, RFC 2997, November 2000.

[27] Bernet, Y. et al., A Framework for Integrated Services Operation over Diffserv Networks, RFC 2998, November 2000.

[28] Black, D. et al., An Architecture for Differentiated Services, RFC 2475, December 1998.

[29] Bless, R., Carpenter, B., Nichols, K. & Wehrle, K., A Lower Effort Per-Domain Behavior for Differentiated Services, Internet Draft, <draft-bless-diffserv-pdb-le-01.txt>, work in progress, November 2002.

[30] Braden, R. et al., Resource Reservation Protocol (RSVP) – Version 1 Functional Specification, RFC 2205, September 1997.

[31] Braden, R., Clark, D. & Shenker, S., Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994.

[32] Braden, R., et al, Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309, April 1998.

[33] Brunner, M. et al., Requirements for Signaling Protocols, Internet Draft, <draft-ietf-nsis-req-06.txt>, work in progress, December 2002.

[34] CADENUS Project, http://www.cadenus.org, last visited at 21/01/2003.

[35] CAIDA Tools Site, http://www.caida.org/tools, last visited at 21/01/2003.

[36] Campanella, M., et al., Specification and Implementation plan for a Premium IP service, GÉANT deliverable D9.1, March 2001.

[37] Campanella, M., Chivalier P., Sevasti, A. & Simar, N., Quality of Service Definition, SEQUIN Deliverable D2.1, March 2001.

[38] Carpenter, B., Architectural Principles of the Internet, RFC 1958, June 1996.

[39] Carpenter, B. & Brim, S., Middleboxes: Taxonomy and Issues, RFC 3234, February 2002.

[40] Chalmers, D. & Sloman, M., A Survey of Quality of Service in Mobile Computing Environments, *IEEE Communications Surveys*, **2** (2), pp. 2-10, 2nd Quarter 1999.

[41] Charny, A. & Le Boudec, J.-Y., Delay Bounds in a Network With Aggregate Scheduling, in Proceedings of the 1st International Workshop on Quality of future Internet Services (QofIS'2000), Berlin, Germany, pp. 1-13, September 2000.

[42] Cheng, L., A Framework for Internet Network Engineering, Internet Draft, <draft-cheng-network-engineering-framework-01.txt>, work in progress, July 2001.

[43] Chicago NAP, Multi-Lateral Peering Agreement, http://nap.aads.net/MLPA.html, last visited at 12/08/2002.

[44] Chuah, C. N., et al., QoS Provisioning Using a Clearing House Architecture, in Proceedings of the Eighth IEEE/IFIP International Workshop on Quality of Service (IWQoS '2000), Pittsburgh, USA, pp. 115-124, June 2000.

[45] Chuah, C. N., *A Scalable Framework for IP-Network Resource Provisioning Through Aggregation and Hierarchical Control*, Ph.D. Thesis, Berkeley, University of California at Berkeley, Fall 2001.

[46] Cisco Systems, Cisco IOS Software - Quality of Service Solutions, White Paper, http://www.cisco.com/warp/public/cc/pd/iosw/tech/qosio_wp.htm, August 2000.

[47] Clark. D., The Design Philosophy of the DARPA Internet Protocols, in Proceedings of the ACM SIGCOMM ’88, Stanford, USA, pp. 106-114, August 1988.

[48] Clark, D., Shenker, S., Zhang, L., Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism, in Proceedings of the ACM SIGCOMM´92, Baltimore, USA, pp. 14-26, August 1992.

[49] Clark, D. & Fang, W., Explicit Allocation of Best-Effort Packet Delivery Service, *IEEE Transactions on Networking*, **6** (4), pp. 362-373, August 1998.

[50] Clark, E., SLAs: In Search of the Performance Payoff, *Network Magazine*, http://www.networkmagazine.com/article/NMG20000509S0030, May 1999.

[51] Cortese, G., et al., Mediation components Release 1 – Requirements and Architecture, CADENUS Deliverable D3.1, March 2001.

[52] Cortese, G. et al., CADENUS: Creation and Deployment of End-User Services in Premium IP Networks, *IEEE Communications Magazine*, **41** (1), pp. 54-60, January 2003.

[53] Crawley, E. et al., A Framework for QoS-based Routing in the Internet, RFC 2386, August 1998.

[54] Cristallo, G. & Jacquenet, C, Providing Quality of Service Indication by the BGP-4 Protocol: the QOS_NLRI attribute, Internet Draft, <draft-jacquenet-qos-nlri-05.txt>, work in progress, October 2002.

[55] Croll, A. & Packman, E., *Managing Bandwidth: Deploying QoS in Enterprise Networks*, Prentice Hall, Upper Saddle River, NJ, USA, 1999.

[56] Cselenyi, I., et al., Measurement of Performance Metrics and Service Events, EURESCOM P1008 Deliverable D3.2, May 2001.

[57] Cukier, K. N., Peering and Fearing: ISP Interconnection and Regulatory Issues, in Proceedings of the Conference on the “Impact on Communications Policy”, Cambridge, USA, December 1997.

[58] Cunha, P. & Queiroz, J. A. M., *Sistemas Distribuídos: de Especificações LOTOS a Implementações*, IX Escola de Computação, Recife, Brazil, 1994.

[59] Davie, B. et al., An Expedited Forwarding PHB, RFC 3246, March 2002.

[60] Demichelis, C. & Chimento, P., IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), RFC 3393, November 2002.

[61] De Serres, Y. & Hegarty, L., Value-Added Services in the Converged Network, *IEEE Communications Magazine*, **39** (9), pp. 146-154, September 2001.

[62] Dewan, R., Freimer, M., & Gundepudi, P., Interconnection Agreements between Competing Internet Service Providers, in Proceedings of the 33rd Hawaii International Conference on System Sciences, Honolulu, USA, January 2000.

[63] Doyle, L., Equinix: Data Center Profile, IDC Bulletin, August 2001, http://www.equinix.com, last visited at 21/01/2003.

[64] Duan, Z., Zhang Z. & Hou, T. T., Service overlay Networks: SLAs, QoS and Bandwidth Provisioning, in Proceedings of the 10th IEEE International Conference on Network Protocols, Paris, France, November 2002.

[65] Duffield, N.G., Goyal, P. & Greenberg, A., A Flexible Model for Resource Management in Virtual Private Networks, Proceedings of the ACM SIGCOMM’99, Cambridge, USA, pp. 95-108, September 1999.

[66] Durham, D. et al., The COPS (Common Open Policy Service) Protocol, RFC 2748, January 2000.

[67] Engel, T. et al., AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture, *IEEE Communications Magazine*, **41** (1), pp. 46-53, January 2003.

[68] Equinix, Inc., http://www.equinix.com, last visited at 18/01/2003.

[69] Estrin, D., Postel, J. & Rekhter, Y., Routing Arbiter Architecture, *Connexions Magazine*, **8** (8), pp 2-7, August 1994.

[70] EURESCOM P1008, Inter-Operator Interfaces for Ensuring End to End QoS, http://www.eurescom.de/public/projects/P1000-series/p1008, 2000.

[71] FAPESP, PTT - Ponto de Troca de Tráfego da Rede ANSP, 2001, http://www.ansp.br:8080 /fbr/ptt, last visited at 21/01/2003.

[72] Fang, W., Differentiated Services: Architecture, Mechanisms and an Evaluation, Ph.D. Thesis, Princeton University, November 2000.

[73] Fang, W., Perterson, L., Inter-AS Traffic Patterns and Their Implications, in Proceedings of the IEEE GLOBECOM'99, Rio de Janeiro, Brazil, December 1999.

[74] Fankhauser, G., Schweikert, D., Plattner, B., Service Level Agreement Trading for the Differentiated Services Architecture, Technical Report n^{o} 59, Swiss Federal Institute of Technology, January 2000.

[75] Fasbender, A. et al., Any Network, Any Terminal, Anywhere, *IEEE Wireless Communications*, **6** (2), pp. 22-30, April 1999

[76] Faucher, F. L. et al., Requirements for support of Diff-Serv-aware MPLS Traffic Engineering, Internet Draft, <draft-ietf-tewg-diff-te-reqts-06.txt>, work in progress, September 2002.

[77] Ferguson, P. & Huston, G., Quality of Service in the Internet: Fact, Fiction, or Compromise?, in Proceedings of the Internet Society's 12th Annual INET Conference (INET '98), Geneva, Switzerland, July 1998.

[78] Ferguson, P. & Huston, G., *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, John Wiley & Sons, New York, 1998.

[79] Ferrari, D. & Verma, D., A Scheme for Real-Time channel establishment in wide-area networks, *IEEE Journal on Selected Areas in Communications*, **8** (3), pp. 368-379, April 1990.

[80] Fidalgo, J., Sadok, D., Kelner, J. & Fidalgo, R., A Simple Performance Policy Management Environment, Network Control and Engineering for QoS, Security and Mobility, (Net-Com'2002), Paris, France, October 2002.

[81] Fine, M. et al., Differentiated Services Quality of Service Policy Information Base, Internet Draft, < draft-ietf-diffserv-pib-09.txt>, work in progress, June 2002.

[82] Fineberg, V., A Practical Architecture for Implementing End-to-End QoS in an IP Network, *IEEE Communications Magazine*, **40** (1), pp. 122-130, January 2002.

[83] Floyd, S. & Jacobson, V., Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, **1** (4), pp. 397-413, August 1993.

[84] Floyd, S., & Jacobson, V., Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, **3** (4), pp. 365-386, August 1995.

[85] Frei, C. & Faltings, B., Bandwidth Allocation Planning in Communication Networks, IEEE in Proceedings of the GLOBECOM'99, Rio de Janeiro, December 1999.

[86] Frei, C., Abstraction Techniques for Resource Allocation in Communication Networks, Ph.D. Thesis, Lausanne, Swiss Federal institute of Technology, April 2000.

[87] Gai, S., Dutt, D. G., Elfassy, N. & Bernet, Y., RSVP Proxy, Internet Draft, <draft-ietf-rsvp-proxy-03.txt>, work in progress, March 2002.

[88] Gao, L., On Inferring Autonomous System Relationships in the Internet, *IEEE/ACM Transactions on Networking*, **9** (6), pp.733-745, December 2001.

[89] Gareiss, R., Old Boy's Network, *Network Magazine*, http://www.networkmagazine.com /article/NMG20010918S0001, October 1999.

[90] GÉANT, The pan-European Gigabit Research and Education Network, 2001, http://www.dante.net/geant, last visited at 21/01/2003.

[91] Ghani, N., Dixit, S. & Wang, T.-S., On IP-over-WDM Integration, *IEEE Communications Magazine*, **38** (1), pp. 72-84, September 2000.

[92] Giordano, S., et al., Advanced QoS Provisioning in IP Networks: The European Premium IP Projects, *IEEE Communications Magazine*, **41** (1), pp. 30-36, January 2003.

[93] Goderis, D. et al., Service Level Specification Semantics and Parameters, Internet Draft, <draft-tequila-sls-02.txt>, work in progress, January 2002.

[94] Goderis, D. et al., Functional Architecture Definition and Top Level Design, TEQUILA Deliverable D1.1, September 2000.

[95] Golmie, N., Mouveaux, F. & Su, D., Differentiated Services over Cable Networks, in Proceedings of GLOBECOM'99, Rio de Janeiro, Brazil, pp. 1109-1115, December 1999.

[96] Goodman, B., Internet Telephony and Modem Delay, *IEEE Network Magazine*, **13** (3), pp. 8-16, May/June 1999.

[97] Govindan, R., Alaettinoglu, C., Varadhan, K. & Estrin, D., Route Servers for Inter-Domain Routing, *The International Journal of Computer and Telecommunications Networking*, **30** (12), pp 1157-1174, December 1998.

[98] Grossman. D., New Terminology and Clarifications for Diffserv, RFC 3260, April 2002.

[99] Günter, M. & Braun, T., Evaluation of Bandwidth Broker Signaling, in Proceedings of the 7th IEEE International Conference on Network Protocols, Toronto, Canada, pp. 145-152, October 1999.

[100] Hatch, C. et al., Selected Scenarios and requirements for end-to-end IP QoS management, EURESCOM P1008 Deliverable D2.1, January 2001.

[101] Hawkinson, J. & Bates, T., Guidelines for creation, selection, and registration of an Autonomous System (AS), RFC 1930 (BCP 6), March 1996.

[102] Heinamen, J. et al., Assured Forwarding PHB Group, RFC 2597, June 1999.

[103] Hoffman, U. & Milouchewa, I., Distributed Measurement and Monitoring in IP Networks, in Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2001), Orlando, Florida, July 2001.

[104] Huitema, C., *Routing in the Internet*, Prentice Hall, Upper Saddle River, NJ, USA, 2.ed. 2000.

[105] Huston, G., Interconnection, Peering, and Settlements, in Proceedings of the Internet Society's 13th Annual INET Conference (INET ’99), San Jose, USA, June 1999

[106] Huston, G., *ISP Survival Guide*: *Strategies for Running a Competitive ISP*, J. Wiley, New York, 1998.

[107] IANA, Internet Assigned Numbers Authority, http://www.iana.org, last visited at 21/01/2003.

[108] IETF IPPM Working Group, IP Performance Metrics, November 2001, http://www.ietf.org/html.charters/ippm-charter.html, last visited at 21/01/2003.

[109] IETF ISSLL Working Group, Integrated Services over Specific Link Layers, July 2001, http://www.ietf.org/html.charters/issll-charter.html, last visited at 21/01/2003.

[110] IETF NSIS Working Group, Next Steps in Signaling, March 2002, http://www.ietf.org/html.charters/nsis-charter.html, last visited at 21/01/2003.

[111] Internap Network Services Corporation, http://www.internap.com, last visited at 18/01/2003.

[112] Internap, Bypassing Congested Peering, Internap Network Services, 2001 http://www.internap.com/about/oursolution.html, last visited at 21/01/2003.

[113] Internap, The Internet is a Little Bit Broken, Internap Network Services, 2001 http://www.internap.com/about/theproblem.html, last visited at 21/01/2003.

[114] ITU-T, The investigation report of major standardization activities about QoS of IP services, The Telecommunication Technology Committee (TTC), Working Group 6-5 SWG4, October 2001.

[115] Jacobson, V., Nichols, K. & Poduri, K., The 'Virtual Wire' Per-Domain Behavior, Internet Draft, <draft-ietf-diffserv-pdb-vw-00.txt>, work in progress, July 2000.

[116] Jackowski, S. et al., Integrated Services Mappings for Low Speed Networks, RFC 2688, September 1999.

[117] Jaffe, J. M., Bottleneck Flow Control, *IEEE Transactions on Communications*, **29** (7), pp. 954-962 July 1981.

[118] Jain, R., Myths about Congestion Management in High Speed Networks, *Internetworking: Research and Experience*, **3** (3), pp. 101-113, September 1992.

[119] Jain, R., *The Art of Computer Systems Performance Analysis*, J. Wiley, New York, 1991.

[120] Jain, M. & Dovrolis, C., End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput, in Proceedings of the ACM SIGCOMM, Pittsburgh, USA, August 2002.

[121] Jamin, S., Shenker, S, Danzig, P., Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service, in Proceedings of the IEEE INFOCOM´97, Kobe, Japan, pp. 973-980, April 1997.

[122] Jamin, S., Danzig, P., Shenker, S., and Zhang, L, A Measurement-based Admission Control Algorithm for Integrated Services Packet Network, *IEEE/ACM Transactions in Networking*, **5** (1), pp. 56-70, February 1997.

[123] Jamoussi, B., Constraint-Based LSP Setup using LDP, RFC 3212, January 2002.

[124] Jeong, J., Lee, S., Kim, Y. & Choi, Y., Design and Implementation of One-way IP Performance Measurement Tool, *Lecture Notes in Computer Science*, Springer-Verlag, **2343**, p. 673-686, July 2002.

[125] Kamienski, C.A. & Sadok, D. Strategies for Provisioning End-to-End QoS-based Services in a Multi-Domain Scenario, submitted to the 18th International Teletraffic Congress (ITC'18), 2003.

[126] Kamienski, C.A. & Sadok, D. The Case for Interdomain Dynamic QoS-based Service Negotiation in the Internet, submitted to the Elsevier International Journal for the Computer and Telecommunications Industry.

[127] Kamienski, C.A., et al., Simulando a Internet: Aplicações na pesquisa e no ensino, XXI Jornada de Atualização em Informática (JAI'2002), Florianópolis, Brazil, July 2002.

[128] Kamienski, C.A., Sadok, D. & Frery, A., Justiça em Modelos de Negociação de Serviços na Internet, in Proceedings of the 20th Brazilian Symposium on Computer Networks, Búzios, Brazil, pp. 685-700, May 2002.

[129] Kamienski, C.A. & Sadok, D., Chameleon: an Architecture for Advanced End-to-End Services in the Internet, Proceedings of the Second IEEE Latin American Network Operations and Management Symposium (LANOMS'2001), Belo Horizonte, Brazil, August 2001.

[130] Kamienski, C.A. & Sadok, D., Chameleon: uma Arquitetura para Serviços Avançados Fim a Fim na Internet com QoS, in Proceedings of the 19th Brazilian Symposium on Computer Networks, Florianópolis, Brazil, May 2001.

[131] Kamienski, C.A. & Sadok, D., Engenharia de Tráfego em uma Rede de Serviços Diferenciados, in Proceedings of the 18th Brazilian Symposium on Computer Networks Belo Horizonte, Brazil, May 2000.

[132] Kamienski, C.A. & Sadok, D., Qualidade de Serviço na Internet, minicurso, 18th Brazilian Symposium on Computer Networks Belo Horizonte, Brazil, May 2000.

[133] Kamienski, C.A. & Sadok, D., Service Definition and Negotiation in the Chameleon Architecture, in Proceedings of the First IEEE International Symposium on Telecommunications (IST 2001), Tehran, Iran, September 2001.

[134] Keshav, S., *An Engineering Approach to Computer Networking*, Addison Wesley, Reading, USA, 1997.

[135] Knightly, E. W. & Shroff, N. B., Admission Control for Statistical QoS: Theory and Practice, *IEEE Network Magazine*, **13** (2), pp. 20-29, March/April 1999.

[136] Lanier, J., Tele-Immersion: The Ultimate QoS-Critical Application, in Proceedings of the First Internet2 Joint Applications/Engineering QoS Workshop, Santa Clara, USA, pp. 17-18, May 1998.

[137] Le Faucheur, et al., MPLS Support of Differentiated Services, RFC 3270, May 2002.

[138] Lefelhocz, C. et al., Congestion Control for Best-Effort Service: Why We Need A New Paradigm, *IEEE Network Magazine*, **10** (1), pp. 10-19, January/February 1996.

[139] Leinen, S., Przybylski, M, Reijs, V. & Trocha, S., Testing of Traffic Measurement Tools, TF-NGN Deliverable D9.4, September 2001.

[140] Liebeherr, J., Patek, S. & Yilmaz, E., Tradeoffs in Designing Networks with End-to-End Statistical QoS Guarantees, in Proceedings of the IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS '2000), Pittsburgh, USA, June 2000.

[141] Lin, Y., Yin, W. & Huang, C., An Investigation into HFC MAC Protocols: Mechanisms, Implementation, and Research Issues, *IEEE Communication Surveys*, Third Quarter, 2000, http://www.comsoc.org/pubs/surveys, last visited at 15/01/2003.

[142] Matsumoto, M. & Nishimura, T., Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, *ACM Transactions on Modeling and Computer Simulation*, **8** (1), pp. 3-30, January 1998.

[143] MCI & NSF, very High Performance Backbone Network Service, http://www.vbns.org, last visited at 12/08/2002.

[144] Merit Network, Route Server Technical Overview, Merit Network, 2000, http://www.rsng.net/overview.html, last visited at 22/01/2003.

[145] Merit Network, Overview of the Internet Routing Registry (IRR), Merit Network, 2000 http://www.irr.net/docs/overview.html, last visited at 22/01/2003.

[146] Miller, K. W., Random Number Generators: Good Ones are Hard to Find, *Communications of the ACM*, **31 (**10), pp. 1192-1201, October 1988.

[147] Mykoniati, E., et al., Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach, *IEEE Communications Magazine*, **41** (1), pp. 38-44, January 2003.

[148] Network Simulator (version 2.1b8a), http://www.isi.edu/nsnam/ns/, August 2001.

[149] Nguyen, T. M. T., Boukhatem, N., Doudane, Y. G. & Pujolle, G., COPS-SLS: A Service Level Negotiation Protocol for the Internet, *IEEE Communications Magazine*, **40** (5), pp. 158-165, May 2002.

[150] Nichols, K. et al., Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, December 1998.

[151] Nichols, K., Jacobson, V. & Zhang, L., A Two-bit Differentiated Services Architecture for the Internet, RFC 2638, July 1999.

[152] Nichols, K. & Carpenter, B., Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification, RFC 3086, April 2001.

[153] Nikolouzou, E., Politis, G., Sampatakos, P. & Venieris, I.S., An Adaptive Algorithm for Resource Management in a Differentiated Services Network, IEEE International Conference on Communications (ICC'2001), Helsinki, Finland, June 2001.

[154] Norton, W. B., Interconnection Strategies for ISPs, personal draft v2.0, May 1999, http://www.equinix.com/press/ISPInterconnectionStrategies2.pdf, last visited at 22/01/2003.

[155] Norton, W. B., Internet Service Providers and Peering, personal draft v2.5, July 2001, http://www.equinix.com/press/PeeringWP.2.pdf, last visited at 22/01/2003.

[156] Oetiker, T., Multi Router Traffic Grapher, http://www.mrtg.org, last visited at 16/01/2003.

[157] Odlyzko, A. P., The Current State and Likely Evolution of the Internet, in Proceedings of the IEEE GLOBECOM'99, Rio de Janeiro, Brazil, December 1999.

[158] Open Peering Initiative, Multi-Lateral peering Agreement, http://www.openpeering.nl /registry/mlpa.html, last visited at 21/01/2003.

[159] Pan, P. P., Hahne, E. L. & Schulzrinne, H. G., BGRP: Sink-Tree-Based Aggregation for Inter-Domain Reservations, *IEEE Journal of Communications and Networks*, **2** (2), pp. 157-167, June 2000.

[160] Pan, P., Scalable Resource Reservation Signaling in the Internet, Ph.D. Thesis, New York, Columbia University, 2002.

[161] Paxson, V., Almes, G., Mahdavi J. & Mathis, M., Framework for IP Performance Metrics, RFC 2330, May 1990.

[162] Perros, H. & Elsayed, K., Call Admission Control Schemes: A Review, *IEEE Communications Magazine*, **34** (11), pp. 82-91, November 1996.

[163] Politis, G.A., Sampatakos, P., Venieris, I. S., Design Of A Multi-Layer Bandwidth Broker Architecture, in Proceedings of the Fifth International Symposium on Interworking (Interworking'2000), Bergen, Norway, October 2000.

[164] Potts, M., CADENUS Project presentation, CADENUS Deliverable D5.1, March 2001.

[165] Pras, A., van Beijnum, B., Sprenkels, R. & Parhonyi, R., Internet Accounting, *IEEE Communications Magazine*, **39** (5), pp. 108-113, May 2001.

[166] Priggouris, G., Hadjiefthymiades, S. & Merakos, L., Supporting IP QoS in the General Packet Radio Service, *IEEE Network Magazine*, **14** (5), pp. 8-17, September/October 2000.

[167] Rajan, R., Celenti, E. & Dutta, S., Service Level Specification for Inter-domain QoS Negotiation, Internet Draft, <draf-sls-somefolks-00.txt>, work in progress, November 2000.

[168] Rekhter, Y. & Li, T., A Border Gateway Protocol 4 (BGP-4), RFC 1771, March 1995.

[169] Ricciato, F. et al., Specification of traffic handling for the first trial, AQUILA Deliverable D1.3, July 2000.

[170] RNP, RNP2: Brazilian Internet2 Infrastructure, http://www.rnp.br/rnp2_en, last visited at 22/01/2003.

[171] RNP CEO, Router Statistics in RNP Backbone, http://www.rnp.br/ceo_en/ceo-estatisticas.html, last visited at 22/01/2003.

[172] Rosen, E., Viswanathan, A. & Callon, R., Multiprotocol Label Switching Architecture, RFC 3031, January 2001.

[173] Roth, R., et al., IP QoS Across Multiple Management Domains: Practical Experiences from Pan-European Experiments, *IEEE Communications Magazine*, **41** (1), pp. 62-69, January 2003.

[174] Salsano, S. et al., Inter-domain QoS Signaling: the BGRP Plus Architecture, Internet Draft, <draft-salsano-bgrpp-arch-00.txt>, work in progress, May 2002.

[175] Salsano, S. et al., Definition and usage of SLSs in the AQUILA consortium, Internet Draft, < draft-salsano-aquila-sls-00.txt>, work in progress, November 2000.

[176] Saltzer, J., Reed, D. & Clark, D., End-to-End Arguments in System Design, *ACM Transactions on Computer Systems*, **2** (4), pp. 277-288, November 1984.

[177] Sang, A., Li, A., A Predictability Analysis of Network Traffic, in Proceedings of the IEEE INFOCOM'2000, Tel-Aviv, Israel, 342-351, March 2000.

[178] Semret, N, Liao, R., Campbell, A. T. & Lazar, A. A., Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections, *IEEE Journal on Selected Areas in Communications*, **18** (12), pp. 2499-2513, December 2000.

[179] Sevasti, A. & Campanella, M., Service Level Agreements specification for IP Premium Service, GEANT deliverable D9.1, addendum 2, August 2001.

[180] Shalunov, S., Teitelbaum, B & Zekauskas, M., A One-way Active Measurement Protocol, Internet Draft, <draft-ietf-ippm-owdp-05.txt>, work in progress, August 2002.

[181] Shalunov, S. & Teitelbaum, B., QBone Scavenger Service (QBSS) Definition, Internet2 Technical Report, March 2001, http://qbone.internet2.edu/qbss, last visited at 22/01/2003.

[182] Shenker, S., Partridge, C. & Guerin, R., Specification of Guaranteed Quality of Service, RFC 2212, September 1997.

[183] Shenker, S. & Wroclawski, J., Network Element Service Specification Template, RFC 2216, September 1997.

[184] Smirnov, M. & Einsiedler, H. J., Key QoS Management Issues, Internet Draft, < draft-smirnov-key-qosissues-00.txt>, work in progress, November 2000.

[185] Smirnov, M. et al., SLA Networks in Premium IP, CADENUS Deliverable D1.1, March 2001.

[186] Stardust.com, Inc., QoS Protocols & Architectures, White Paper, July 1999.

[187] Stardust.com, Inc., The Need for QoS, White Paper, July 1999.

[188] Stemm, M. & Katz, R.H. Vertical handoffs in wireless overlay networks, *ACM Mobile Networks and Applications*, **3** (4), pp. 335-350, December 1998.

[189] Subramanian, L., Stoica, I., Balakrishnan, H. & Katz, R. H., OverQoS: Offering QoS using Overlays, First Workshop on Hop Topics in Networks (HotNets-I), October 2002.

[190] Tanenbaum, A. S., Computer Networks, Prentice Hall, Upper Saddle River, USA, 4.ed., 2003.

[191] Tanenbaum, A. S., Modern Operating Systems, Prentice Hall, Upper Saddle River, USA, 2.ed., 2001.

[192] Teitelbaum, B. & Chimento, P., QBone Bandwidth Broker Architecture, Internet2 Draft, June 2000, http://qbone.internet2.edu/bb/bboutline2.html, last visited at 22/01/2003.

[193] Teitelbaum, B., Sikora, J, & Hanss, T., Quality of Service for Internet2, in Proceedings of the First Internet2 Joint Applications/Engineering QoS Workshop, Santa Clara, USA, pp. 5-16, May 1998.

[194] Teitelbaum, B. et al., QBone Architecture (V1.0), Internet2 Draft, August 1999, http://qbone.internet2.edu/arch, last visited at 22/01/2003.

[195] Teitelbaum, B, Internet2 QBone: Building a Testbed for IP Differentiated Services, *IEEE Network Magazine*, **13** (5), pp. 8-16, September/October 1999

[196] Teitelbaum, B., Future Priorities for Internet2 QoS, Internet2 Draft, October 2001, http://qos.internet2.edu/wg/documents-informational/20011002-teitelbaum-qos-futures.pdf, last visited at 22/01/2003.

[197] Telcordia Netsizer, Internet Growth Forecasting Tool, 2002, http://www.netsizer.com, last visited at 12/08/2002.

[198] TEQUILA Project, http://www.ist-tequila.org, last visited at 22/01/2003.

[199] Terzis, A., et al., A Two-Tier Resource Management Model for the Internet, in Proceedings of the Global Internet Symposium, Rio de Janeiro, Brazil, December 1999.

[200] Terzis, A., A Two-Tier Resource Allocation Framework for the Internet, Ph.D. Thesis, University of California at Los Angeles, 2000.

[201] T'Joens, Y., et al., Service Level Specification and Usage Framework, Internet Draft, <draft-manyfolks-sls-framework-00.txt>, work in progress, October 2000.

[202] Traina, P., McPherson, D. & Scudder, J., Autonomous System Confederations for BGP, RFC 3065, February 2001.

[203] Trimintzios, P. et al., A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks, *IEEE Communications Magazine*, **39** (5), pp. 80-88 May 2001.

[204] Trimintzios, P. et al., Engineering the Multi-Service Internet: MPLS and IP-based Techniques, in Proceedings of the IEEE International Conference on Telecommunications (ICT 2001), Bucharest, Romania June 2001.

[205] Trimintzios, P. et al., Policy-based Network Dimensioning for IP Differentiated Services Networks, in Proceedings of the IEEE Workshop on IP Operations and Management (IPOM 2002), Dallas, USA, November 2002.

[206] Tsetsekas, C., Maniatis, S. & Venieris, I. S., A Middleware Solution for the Support of QoS for Legacy Applications, in Proceedings of the International Conference on Software, Telecommunications and Computer Networks (SOFTCOM 2000), Rijeka, Croatia & Trieste, Italy, October 2000.

[207] Tsetsekas, C., Maniatis, S. & Venieris, I. S., Supporting QoS for Legacy Applications, IEEE International Conference on Networking, (ICN 2001), Colmar, France, July 2001.

[208] UCAID, Abilene Project, http://www.internet2.edu/abilene, last visited at 22/01/2003.

[209] UMTS Forum, Enabling UMTS / Third Generation Services and Applications, Report nº 11, October 2000.

[210] Van der Waaij, BD., et al., Quality-based Service Management, Technical Report, KPN Research (Holland), Project Internet Next Generation, December 1999.

[211] Van Heuven, P. et al., Intermediate-Results based Protocol and Algorithm Specification, TEQUILA Deliverable D1.3, October 2001.

[212] Verma, D., Supporting Service Level Agreements on IP Networks, Macmillan, Indianapolis, USA, 1999.

[213] Virtela Communications, http://www.virtela.com, last visited at 18/01/2003.

[214] Wang, X., & Schulzrinne, H., A Resource Negotiation and Pricing Protocol, in Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'99), Basking Ridge, USA, pp. 77-93, January 1999.

[215] Wang, Z. & Crowcroft, J., Quality of Service Routing for Supporting Multimedia Applications, *IEEE Journal on Selected Areas in Communications*, **14** (7), pp. 1228-1234, September 1996.

[216] Westerinen, A. et al., Terminology for Policy-Based Management, RFC 3198, November 2001.

[217] Wroclawski, J., Specification of the Controlled-Load Network Element Service, RFC 2211, September 1997.

[218] Xiao, X. & Ni, L. M., Internet QoS: A Big Picture, *IEEE Network Magazine*, **13** (2), pp. 8-18, March/April 1999.

[219] Xiao, X., Hannan, A., Bailey, B. & Ni, L. M., Traffic Engineering with MPLS in the Internet, *IEEE Network Magazine*, **14** (2), pp. 28-33, March/April 2000.

[220] Xiao, X., Providing Quality of Service in the Internet, Ph.D. Thesis, Michigan State University, May 2000.

[221] Zhang, Z, Duan, Z. & Hou, Y. T., On Scalable Design of Bandwidth Brokers, *IEICE Transaction on Communications*, **E84-B** (8), pp.2011-2025, August 2001.