

# **A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks**

P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, Univ. of Surrey, UK

D. Griffin, University College London, UK

P. Georgatsos, Algonet S.A., Greece

D. Goderis, Y. T'Joens, Alcatel, Belgium

L. Georgiadis, Aristotle Univ. of Thessaloniki, Greece

C. Jacquenet, France Telecom R&D, France

R. Egan, Racal Research, UK

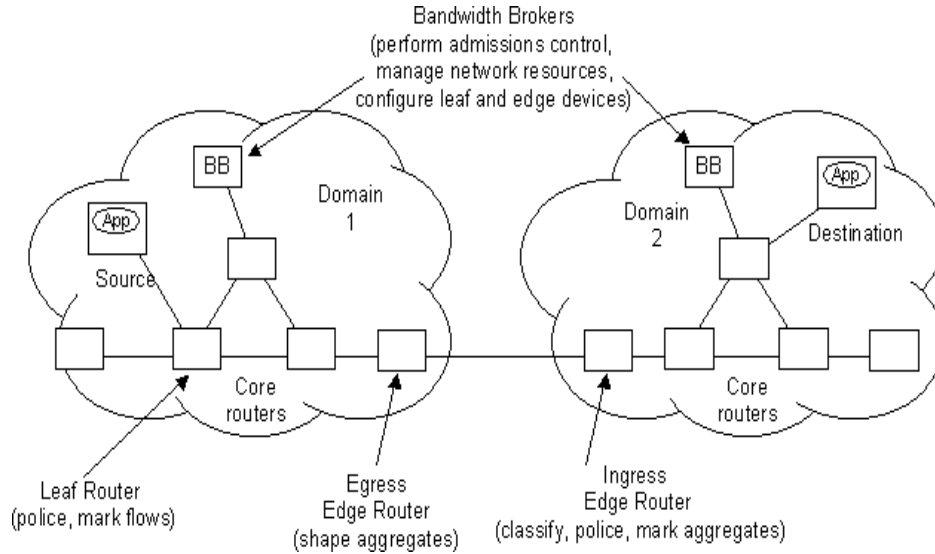
## **Abstract**

As the Internet evolves towards the global multi-service network of the future, a key consideration is support for services with guaranteed Quality of Service (QoS). The proposed Differentiated Services (DiffServ) framework is seen as the key technology to achieve this. DiffServ currently concentrates on control/data plane mechanisms to support QoS but also recognises the need for management plane aspects through the Bandwidth Broker (BB). In this paper we propose a model and architectural framework for supporting DiffServ-based end-to-end QoS in the Internet, assuming underlying MPLS-based explicit routed paths. The proposed integrated management and control architecture will allow providers to offer both quantitative and qualitative based services while optimising the use of underlying network resources.

## **1 Introduction**

With the prospect of becoming the ubiquitous all-service network of the future, the Internet needs to evolve to support services with guaranteed QoS characteristics. The Internet Engineering Task Force (IETF) has proposed a number of QoS models and supporting technologies including the Integrated (IntServ) and DiffServ [RFC-2475] frameworks. The latter has been conceived to provide QoS in a scalable fashion. Instead of maintaining per-flow soft state at each router, packets are classified, marked and policed at the edge of a DiffServ domain. A limited set of Per Hop Behaviours (PHBs) differentiate the treatment of aggregate flows in the core of the network, in terms of scheduling priority, forwarding capacity and buffering. Service Level Specifications (SLs) are used to describe the appropriate QoS parameters the DiffServ-aware routers will have to take into account, when enforcing a given PHB. Thus micro-flow-based treatment is restricted at the DiffServ domain border while the transit routers deal only with aggregate flows, according to the Differentiated Services Code-Point (DSCP) field of the IP header. This procedure leads to the provision of coarse-grained QoS to applications in a qualitative instead of a quantitative fashion, although quantitative QoS guarantees could also be provided using for example the Expedited Forwarding (EF) PHB.

In order to achieve such QoS guarantees, control plane mechanisms are used to reserve resources on demand but management plane mechanisms are also used to plan and provision the network and to manage requirements for service subscription according to available resources [GEORG99]. QoS frameworks such as IntServ and DiffServ have so far concentrated in control plane mechanisms for providing QoS. However, it would not seem possible to provide QoS without the network and service management support, which is an integral part of QoS-based telecommunication networks. Considering in particular the DiffServ architecture (see Figure 1), a key issue is end-to-end QoS delivery. The DiffServ architecture suggests only mechanisms for relative packet forwarding treatment to aggregate flows, traffic management and conditioning; by no means does it suggest an architecture for end-to-end QoS delivery. In order to provide end-to-end quantitative QoS guarantees, DiffServ mechanisms should be augmented with intelligent traffic engineering functions.



**Figure 1 The DiffServ architecture.**

Traffic Engineering (TE) is in general the process of specifying the manner in which traffic is treated within a given network. TE has both user and system-oriented objectives. The users expect certain performance from the network, which in turn should attempt to satisfy these expectations. The expected performance depends on the type of traffic that the network carries, and is specified in the SLS contract between customer and Internet Service Provider (ISP). The network operator on the other hand should attempt to satisfy the user traffic requirements in a cost-effective manner. Hence, the target is to accommodate as many as possible of the traffic requests by using optimally the available network resources. Both objectives are difficult to realise in a multi-service network environment.

Multi-Protocol Label Switching (MPLS) [ROSE00], is an important emerging technology for enhancing IP in both features and services. Although, the concept of TE does not depend on specific layer 2 technologies, it is argued that MPLS [AWDU00] is the most suitable tool to provide it. MPLS allows sophisticated routing control capabilities as well as QoS resource management techniques to be introduced to IP networks. With the advent of Differentiated Services and MPLS, IP traffic engineering has attracted a lot of attention in recent years. [AUKI00], [FELD00], [QBONE], are a few of the most recent projects in this area. The TEQUILA project (Traffic Engineering for Quality of Service in the Internet, at Large Scale)<sup>1</sup> is one of them. The objective of TEQUILA is to study, specify, implement and validate a set of service definition and traffic engineering tools in order to obtain quantitative end-to-end QoS guarantees through careful dimensioning, admission control and dynamic resource management of DiffServ networks.

This paper discusses issues in this area and proposes an architectural framework for end-to-end QoS in the Internet. We take the position that the future Internet should offer a *variety* of service quality levels ranging from those with explicit, hard performance guarantees for bandwidth, loss and delay characteristics down to low-cost services based on best-effort traffic, with a range of services receiving qualitative traffic assurances occupying the middle ground. Assuming a DiffServ MPLS IP-based network infrastructure, we propose a functional architecture for TE specifying the required components and their interactions for end-to-end QoS delivery. The starting point is the specification of SLSs agreed between ISPs and their customers, and their peers, with confidence that these agreements can be met. The SLSs reflect the elemental QoS-based services that can be offered and supported by an ISP and set the objectives of the TE functions, these being fulfilment and assurance of the SLSs. The proposed framework ensures that agreed SLSs are adequately provisioned and that

<sup>1</sup> See: <http://www.ist-tequila.org/>

future SLSs may be negotiated and delivered through a combination of static, quasi-static and dynamic traffic engineering techniques both *intra*- and *inter*-domain. It proposes solutions for operating networks in an optimal fashion through planning and dimensioning and subsequently through dynamic operations and management functions (“*first plan, then take care*”).

## 2 Service Level Specifications

In this section we substantiate the notion of Service Level Specification [RFC-2475]. The definition of SLSs is the first step towards the provisioning of QoS. Today, QoS-based services are offered in terms of contract agreements between an ISP and its customers. Such agreements, and especially the negotiations preceding them, will be greatly simplified through a standardised set of SLS parameters. A SLS standard is also necessary to allow for a highly developed level of automation and dynamic negotiation of SLSs between customers and providers. Moreover, the design and the deployment of BB capabilities [RFC-2638] require a standardised set of semantics for SLSs being negotiated between both the customer and ISP and among ISPs.

Note that although we allow for a number of performance and reliability parameters to be specified, in practice a provider would only offer a finite number of services, even for those with quantitative QoS guarantees. Therefore, parameters such as delay, mean-down-time, etc. could only take discrete values from the set offered by a particular provider. While offering customers a well-defined set of service offerings, this approach simplifies the TE problem from the providers’ perspective.

### 2.1 Contents and Semantics

The contents of a SLS [GODE00b] include the essential QoS-related parameters, including scope and flow identification, traffic conformance parameters and service guarantees. More specifically a SLS has the following fields: Scope, Flow Description, Traffic Conformance Testing, Excess Treatment, Performance Parameters, Service Schedule and Reliability.

The *Scope* of an SLS associated to a given service offering uniquely identifies the geographical and topological region over which the QoS of the IP service is to be enforced. An ingress (or egress) interface identifier should uniquely determine the boundary link or links as defined in [RFC-2475] on which packets arrive/depart at the border of a DS domain. This identifier may be an IP address, but it may also be determined by a layer-two identifier in case of e.g. Ethernet, or for unnumbered links like in e.g., PPP-access configurations. The semantics allow for the description of one-to-one (pipe), one-to-many (hose) and many-to-one (funnel) communication SLS-models, denoted respectively by (1|1), (1|N) and (N|1).

The *Flow Description* (*FlowDes*) of an SLS associated to a given service offering indicates for which IP packets the QoS policy for that specific service offering is to be enforced. A SLS has only one *FlowDes*, which can be formally specified by providing one or more of the following attributes:

FlowDes = (DiffServ information, source information, destination information, application information)

Setting one or more of the above attributes formally specifies a SLS *FlowDes*. The DiffServ information might be the DSCP. The source/destination information could be a source/destination address, a set of them, a set of prefixes or any combination of them. The *FlowDes* provides the necessary information for classifying the packets at a DiffServ edge node. The packet classification can either be Behaviour Aggregate (BA) or Multi-Field (MF) based.

*Traffic Envelope and Traffic Conformance* describes the traffic characteristics of the IP packet stream identified by *FlowDes*. The traffic envelope is a set of Traffic Conformance (TC) parameters, describing how the packet stream should be in order to receive the treatment indicated by the *Performance Parameters* (see below). The TC parameters are the input to the *Traffic Conformance Testing* algorithms. The traffic conformance testing is the set of actions, which uniquely identifies the

“in-profile” and “out-of profile”<sup>2</sup> (or excess) packets of an IP stream identified by the FlowDes. The TC Parameters describe the reference values the traffic identified by the FlowDes will have to comply with. The TC Algorithm is the mechanism enabling to unambiguously identify all “in” or “out” of profile packets based on these conformance parameters. The following is a non-exhaustive list of potential conformance parameters: *peak rate*  $p$  in bits per sec (bps), *token bucket rate*  $r$  (bps), *bucket depth*  $b$  (bytes), *minimum MTU* - Maximum Transfer Unit -  $m$  (bytes) and *maximum MTU*  $M$  (bytes).

An *Excess Treatment* parameter describes how the service provider will process excess traffic, i.e. out-of-profile traffic (or other than the in-profile in the case of multi-level TC). The process takes place after Traffic Conformance Testing. Excess traffic may be dropped, shaped and/or remarked. Depending on the particular treatment, more parameters may be required, e.g. the DSCP value in case of re-marking or the shapers buffer size for shaping.

The *Performance Parameters* describe the service guarantees the network offers to the customer for the packet stream described by the FlowDes and over the geographical/topological extent given by the scope. There are four performance parameters: *delay*, *jitter*, *packet loss*, and *throughput*.<sup>3</sup> *Delay* and *jitter* indicate respectively the maximum packet transfer delay and packet transfer delay variation from ingress to egress. Delay and jitter may either be specified as worst-case (deterministic) bounds or as quantiles. The *packet loss* indicates the loss probability for in-profile packets from ingress to egress. Delay, jitter and packet loss apply only to in-profile traffic. *Throughput* is the rate measured at the egress.

**Table 1 Example SLS parameter settings for various services.**

	Virtual Leased Line Service	Bandwidth Pipe for Data Services	Minimum Rate Guaranteed Service	Qualitative Olympic Services		The Funnel Service
<b>Comments</b>	Example of a uni-directional VLL, with quantitative guarantees	Service with only strict throughput guarantee. TC and ET are not defined but the operator might define one to use for protection.	It could be used for a bulk of ftp traffic, or adaptive video with min throughput requirements	They are meant to qualitatively differentiate between applications such as: on-line web-browsing      e-mail traffic		It is primarily a protection service; it restricts the amount of traffic entering a customer's network
<b>Scope</b>	(1 1)	(1 1)	(1 1)	(1 1) or (1 N)		(N 1) or (all 1)
<b>Flow Description</b>	EF, S-D IP-A	S-D IP-A	AF1x	MBI		AF1x
<b>Traffic Conformance</b>	(b, r) e.g. r=1	NA	(b, r)	(b, r), r indicates a minimum committed Olympic rate		(b, r)
<b>Excess Treatment</b>	Dropping	NA	Remarking	Remarking		Dropping
<b>Performance Parameters</b>	D=20 (t=5, q=10e-3), L=0 (i.e. R = r)	R = 1	R = r	D=low L=low (gold/green)	D=med L=low (silver/green)	NA
<b>Service Schedule</b>	MBI, e.g. daily 9:00-17:00	MBI	MBI	MBI	MBI	MBI
<b>Reliability</b>	MBI, e.g. MDT = 2 days	MBI	MBI	MBI	MBI	MBI

(b, r): token bucket depth and rate (Mbps), p: peak rate, D: delay (ms), L: loss probability, R: throughput (Mbps), t: time interval (min), q: quantile, S-D: Source & Destination, IP-A: IP Address, MBI: May Be Indicated, NA: Not Applicable, MDT: Maximum Down Time (per year), ET: Excess Treatment, TC: Traffic Conformance

<sup>2</sup> Note that the conformance result might not necessarily be of a binary mode (in/out) but it could also be multi-level (e.g. using a Two-rate Three-colour Marker algorithm).

<sup>3</sup> For each of these parameters we must specify a *time interval* and in some cases (e.g. delay) a quantile.

Performance parameters might be either quantitative or qualitative. A performance parameter is quantifiably guaranteed if an upper bound is specified. The service guarantee offered by the SLS is quantitative if at least one of the four performance parameters is quantified. If none of the SLS performance parameters is quantified, then the performance parameters *delay* and *packet loss* may be “qualified”. Possible qualitative values for delay and/or loss are *high*, *medium*, *low*. The actual “quantification” of the relative difference between high, medium and low is a policy-based decision (e.g. high = 2 x medium; medium = 3 x low). If the performance parameters are not quantified nor qualified the service will be best effort.

The *Service Schedule* indicates the start time and end time of the service, i.e. when is the service available. This might be expressed as a collection of the following parameters: time of the day range, day of the week range, and month of the year range. *Reliability* indicates the maximum allowed mean downtime per year (MDT) and the maximum allowed time to repair (TTR) in case of service breakdown. Other parameters might be also included in the SLS for example the *Assurance Level*, which describes the percentage of the time by which the ISP will be able to conform to the other SLS parameters.

### 3 An Architecture for Supporting QoS

In order to support end-to-end QoS based on the SLSs described above, we propose the functional architecture shown in Figure 2 [GODE00a]. There are three main parts in this architecture: SLS Management (SLSM), Traffic Engineering (TE) and Policy Management (PM), in addition to Monitoring and Data Plane functionalities. The SLSM part is responsible for subscribing and negotiating SLSs with users or other peer Autonomous Systems (ASs) and it performs admission control for the dynamic invocation of subscribed SLSs. This part is also responsible for transforming the SLS specific information into aggregate traffic demand (traffic matrix), in order to *feed* the TE part with the necessary input. The TE part is responsible for selecting paths which are capable of meeting the QoS requirements for a given traffic demand. Such information is conveyed between the customer and the service provider during SLS negotiation then it is processed by the Traffic Forecast and is transformed into the aggregate traffic matrix. The TE part of the architecture is responsible for dimensioning the network according to the projected demands, and for establishing and dynamically maintaining the network configuration that has been selected to meet the SLS demand according to the QoS dynamic information provided by the SLSM.

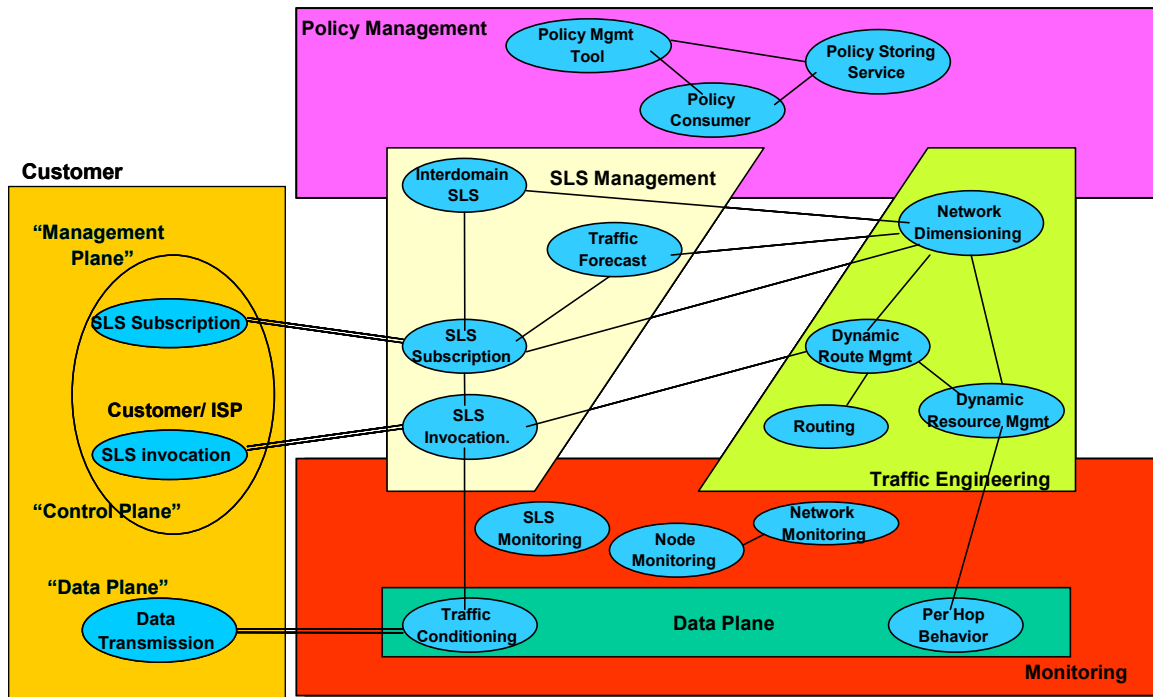


Figure 2 The TEQUILA functional architecture.

## 4 SLS Management

SLS Management is responsible for all SLS-related activities and is further decomposed into four Functional Blocks (FBs): SLS Subscription, SLS Invocation, Traffic Forecast and Inter-Domain SLS Requestor. Figure 2 shows the Interaction of the SLSM component with external customers or ISPs.

*SLS Subscription* (SLS-S) is the FB, which includes processes of customer registration and long-term policy-based admission. The customer might either be a peer Autonomous System (AS) or a business or residential user. The subscription (or registration) concerns the Service Level Agreement (SLA), containing amongst other prices, terms and conditions and the technical parameters of the SLS. The subscription should provide the required *authentication information*. SLS-S contains an SLS repository with the current (long-term) subscriptions and a SLS history repository. This information serves as basic input for the Traffic Forecast. SLS-S performs static “admission control” in the sense that it knows whether a requested long-term SLS can be supported or not in the network given the current network configuration; this is not an instantaneous snapshot of load/spare capacity, but the longer-term configuration provided by Network Dimensioning (described below). It provides a view of the current available resources to the SLS-I FB.

The contract (SLS) subscription constrains the customer's future usage pattern but at the same time guarantees a certain level of performance for invocations conforming to the agreement. This is of benefit to the network operator who can use the information declared in the contract for network dimensioning and TE purposes. It is also of benefit to the customer as it provides a guarantee that network resources will be available when required.

*SLS Invocation* (SLS-I) is the FB, which includes the process of dynamically dealing with a flow and it is part of *control plane* functionality. It performs dynamical admission control as requested by the user and this process can be flow-based. SLS-I receives input from the SLS-S, e.g. for authentication purposes, and has a view on the current spare resources. Admission control is mostly measurement-based and takes place at the network edges. Finally, SLS-I delegates the necessary rules to the traffic conditioner. The rules when enforced will ensure that packets are marked with the correct DSCP, so that out-of-profile packets are handled in a certain way, etc. Both the SLS-S and SLS-I interact with the *Inter-domain SLS Requester*, which deals with all inter-domain SLS *negotiations, subscriptions* and *invocations*. It handles requests for changing/renegotiating the SLSs with the peer ISPs/ASs.

The main function of *Traffic Forecast* (TF) is to generate a traffic estimation matrix to be used by the TE. TF is the “glue” between the SLSM Customer-oriented Framework and the TE Resource-oriented Framework of our functional architecture. The *input* of TF is *SLS (customer) aware* while the *output* is only *Class of Service (CoS) aware*. The *traffic estimation matrix* contains *per CoS type*, the (long-term) estimated traffic that flows between each ingress/egress pair. Its calculation is based on the SLS subscription repository, traffic projections and historical data provided by Monitoring, network physical topology, the physical nature and capacities of the access links, business policies, economic models, etc.

## 5 Traffic Engineering

In general, there exist two TE approaches:

- **MPLS-based TE:** This approach relies on an explicitly routed paradigm, whereby a set of routes (paths) is computed off-line for specific types of traffic. In addition, appropriate network resources (e.g. bandwidth) may be provisioned along the routes according to predicted traffic requirements. Traffic is dynamically routed within the established sets of routes according to network state.
- **IP-based TE:** This approach relies on a ‘liberal’ routing strategy, whereby routes are computed in a distributed manner, as discovered by the routers themselves. Although route selection is performed in a distributed fashion, the QoS-based routing decisions are constrained according to network-wide TE considerations made by the dimensioning and dynamic routing algorithms. The latter dynamically assigns cost metrics to each network interface. Route computation is usually based on shortest or widest path algorithms with respect to the assigned link costs. In order to

allow for routes to be computed per traffic type or class, a link may be allocated multiple costs, one per DSCP.

In this paper we are considering only the MPLS-based approach, although our architecture is independent of particular TE approach, i.e. it can be used to accommodate also pure IP-based TE solutions. The TEQUILA project is studying IP-based TE solutions but these are outside the scope of this paper.

MPLS TE is exercised at two time scales, long-term and short-term.

- *Long-term MPLS TE* (days - weeks) selects the traffic that will be routed by MPLS based on predicted traffic loads and existing long-term SLS contracts. The Explicitly Routed Paths (ERPs) as well as associated router scheduling and buffer mechanisms are defined. This process is done off-line taking into account global network conditions and traffic load. It involves the global trade-offs of user and system-oriented objectives.
- *Short-term MPLS TE* (minutes - hours) is based on the observed state of the operational network. Dynamic resource and route management procedures are employed in order to ensure high resource utilisation and to balance the network traffic across the ERPs specified by long term TE. These dynamic management procedures perform adaptation to current network state within the bounds determined by long-term traffic engineering. Triggered by inability to adapt appropriately, by significant changes in expected traffic load, or by local changes in network topology, ERPs may be created or torn down by long-term TE functions.

The long-term TE corresponds to the *time-based* capacity management functions of TE [AWDU00], whereas short-term TE corresponds to *state-dependent* capacity management functions of TE. By virtue of our model, these functions inter-operate towards a complete TE solution.

## 5.1 Network Dimensioning

Network Dimensioning (ND) is responsible for mapping the traffic onto the physical network resources and configures the network in order to accommodate the forecasted traffic demands. ND defines ERPs (MPLS Label Switched Paths - LSPs) in order to accommodate the expected traffic. The TF FB provides the forecasted demand and ND is responsible for determining cost-effective allocation of physical network resources subject to resource restrictions, load trends, requirements of QoS and policy directives and constraints. The resources that need to be allocated are mainly QoS routing constraints, like link capacities and router buffer space, while the means for allocating these resources are capacity allocation, routing mechanisms, scheduling, and buffer management schemes. The ND component is centralised for a particular AS, although distributed implementations on a sub-domain or area of an AS are also possible. In any case, it utilises network-wide information, received from the network routers and/or other functional components through polling and/or asynchronous events.

The main task of ND, which operates in order of days to weeks<sup>4</sup>, is to accept input about the forecasted demand from TF, and by knowing the physical topology to calculate, in a policy-driven fashion, and install parameters required by the elementary TE functions in the IP routers of the network. The output of ND is the set of ERPs and their associated parameters. The objective of such a calculation is to accommodate all the expected demand, and therefore meet the SLS performance requirements, without overloading any part of the network. This objective leaves space for unpredictable traffic fluctuations (handled by DRtM and DRsM) and at the same time not having to reroute large amounts of traffic in the case of failures. One can devise ND algorithms either in the form of an optimisation problem and use relaxation techniques to overcome the complexity problems or using heuristics. The definition, analysis and testing of such ND algorithms is part of the ongoing work within the TEQUILA project.

---

<sup>4</sup> By which we mean that it is invoked at approximately these intervals - not that the algorithms take this long to converge.

The output of ND is fed to DRtM and DRsM, and also to the SLS Management part of the architecture in order to base the admission control decisions for future SLS subscriptions. Admission control for SLS invocations is based on the information from ND, DRtM and DRsM, with the latter two being more important since they have more up-to-date dynamic information.

## 5.2 Dynamic Route Management

Dynamic Route Management (DRtM) is responsible for managing the routing processes in the network according to the guidelines produced by ND on routing traffic according to QoS requirements associated to such traffic (contracted SLSs).

This FB is responsible mainly for managing the parameters based on which the selection of one of the established LSPs is effected in the network, with the purpose of load balancing. It receives as input the set of ERPs (multiple ERPs per source-destination pair) defined by ND and relies on appropriate network state updates distributed by the DRsM FB. In addition, it informs ND, by sending notifications, on over-utilisation of the defined paths so that appropriate actions are taken (e.g. creation of new paths). In this approach, the functionality of the DRtM is distributed at the network border routers/edges.

In MPLS-based TE the LSP bandwidth is *implicitly* allocated through link scheduling parameters along the topology of the LSPs, while traffic conditioning enforced at an ingress router is used to ensure that input traffic is within its defined capacity.

## 5.3 Dynamic Resource Management

Dynamic Resource Management (DRsM) has distributed functionality, with an instance attached to each router. This component aims at ensuring that link capacity is appropriately distributed between the PHBs sharing the link. It does this by setting buffer and scheduling parameters according to ND directives, constraints and rules and taking into account actual experienced load as compared to required (predicted) resources. Additionally DRsM attempts to resolve any resource contention that may be experienced while enforcing different PHBs. It does this at a higher level than the scheduling algorithms located in the routers themselves.

DRsM gets estimates of the *required* resources for each PHB from ND, and it is allowed to dynamically manage resource reservations within certain constraints, which are also defined by ND. For example, the constraints may indicate the *effective* resources required to accommodate a certain quantity of unexpected dynamic SLS invocations. Compared to ND, DRsM operates on a relatively short time-scale. DRsM manages two main resources: Link Bandwidth and Buffer Space.

*Link Bandwidth:* ND determines the bandwidth required on a link to meet the QoS requirements conveyed in the SLS. DRsM translates this information into scheduling parameters, which are then used to configure link schedulers in the routers. These parameters are subsequently managed dynamically, according to actual load conditions, to resolve conflicts for physical link bandwidth and avoid starving of such bandwidth for the enforcement of some PHBs.

*Buffer Space:* Appropriate management of the buffer space allows packet loss probabilities to be controlled. The buffers also provide a bound on the largest delay that successfully transmitted packets may experience. Buffer allocation schemes in the router dictate how buffer space is split between contending flows and when packets are dropped. According to the constraints imposed by ND for the QoS parameters associated with the traffic of a given PHB, DRsM sets the buffer space and determines the rules for packet dropping in the routers. The drop levels need to be managed as the traffic mix and volume changes. For example, altering the bandwidth allocated to a LSP may alter the bandwidth allocated for the correct enforcement of a corresponding PHB. If the loss probability for the PHB is to remain constant, then the allocated buffer space may need to change.

Through the activities of DRsM, the load-dependent metrics associated with links may change if the metrics do not reflect load directly. For example, a metric defining available free capacity in a PHB rather than used bandwidth may change when scheduling priority is increased for that PHB. For these reasons DRsM issues DRtM with appropriate updates on the state of the allocated resources for PHBs

to be utilised for routing purposes. DRsM also triggers ND when network/traffic conditions are such that its algorithms are no longer able to operate effectively. For example, link partitioning is causing lower priority/best effort traffic to be throttled due to excessive high priority traffic and these conditions cannot be resolved within the constraints previously defined by ND.

## 6 Policy Management

Policy Management includes functions such as the Policy Management Tool (PMT), the Policy Storing Service (PSS) and the Policy Consumers (PCs) or Policy Decision Points (PDPs). The latter correspond to their associated functional blocks, e.g. SLS related admission policies for the SLS Management, dimensioning policies for the ND, dynamic resource/route management policies for the DRsM/DRtM, etc.

Although Figure 2 has shown a single PC/PDP for illustrative purposes, our model assumes many instances of policy consumers [FLEG01]. In reality, the PC/PDP is not a separate component but it is collocated with other functional blocks, e.g. SLS-S & SLS-I, TF, ND, DRtM & DRsM. Targets can be the managed objects of the associated FB or of lower-level FBs. PCs need also to have direct communication with the Monitoring FB in order to get information about traffic-based policy-triggering events. Note that triggering events may be also other than traffic-related.

Policies are defined in the PMT using a high-level language, and are then translated to object-oriented policy representation (information objects) and stored in the policy repository, i.e. PSS. New policies are checked for conflicts with existing policies, although some conflicts may only be detected at run time. After the policies are stored, activation information may be passed to the associated PC/PDP.

Every time the operator introduces a high level policy, this should be refined into policies for each layer of the TEQUILA functional architecture forming a policy hierarchy that reflects the management hierarchy [FLEG01]. The administrator should define generic classes of policies and provide some refinement logic/rules for the policy classes that will help the automated decomposition of instances of these classes into policies for each level of the hierarchical management system shown in Figure 2.

## 7 Working System Scenario

In this section, we will describe a working scenario and the information flow of the functional architecture that was presented in the previous sections.

Let's assume that several customers are attached to an AS which employs the TEQUILA System. These customers are negotiating SLSs with the SLS-S FB. Let's assume that at some point in time there are  $N$  subscribed SLSs, and at time  $t$  re-dimensioning needs to be done. The reasons for re-dimensioning might be: (a) the amount of spare resources for future SLS subscriptions is below a (policy defined) threshold, (b) the amount of the SLS subscription rejections is greater than a (policy-based) defined threshold, (c) DRtM or DRsM are unable to handle the current resource demand, (d) the re-dimensioning cycle has elapsed (dimensioning period).

First, ND will request the traffic forecast (matrix) that corresponds to the next dimensioning period. TF will consider the currently subscribed  $N$  SLSs, the (policy-based) additional  $M$  SLS subscription requests that are predicted for the next dimensioning period, the (policy-based) over-subscription ratio, and historical monitoring data, in order to prepare the traffic matrix<sup>5</sup>. The demand provided to ND will be something between  $(N, N+M)$ . ND will run some optimisation or heuristic dimensioning algorithm in order to define multiple paths (trees) between the ingress and (list of) egress nodes, i.e. the configuration of the network for the next dimensioning period. ND needs to provide this configuration information back to SLS-S in order to be able to perform admission control at the level of subscriptions. This information is also passed to DRtM and DRsM, which contact the Network

---

<sup>5</sup> There are actually more than one such matrices, one per Class of Service (CoS), where the CoS might be defined by the: (a) Ordered Aggregate (OA) [RFC-2475], and (b) a range of delay. E.g. CoS1 = [EF, 1-1.5ms]. Also because we assume various scopes (pipe, hose, funnel), the most appropriate data structure might not be a matrix, but a linked list.

Elements (NEs) in order to enforce this configuration by setting LSPs and configuring the various PHBs. Finally, Monitoring needs to be informed about this configuration in order to set the appropriate monitoring engines.

The SLS-S will use the configuration received from ND to decide for future subscriptions but will also pass it to SLS-I in order for it to have the necessary information for invocation admission control. Now let's assume that several SLSs are being invoked. For each of these SLSs the SLS-I will check the SLS repository to see if it corresponds to a subscribed customer. The current load information (taken from monitoring) will also be checked against the current network configuration in order to decide whether a particular SLS can be accepted or not. If it is accepted, SLS-I will configure the Traffic Conditioners (Data Plane) appropriately. When the actual traffic arrives, DRtM will balance the load among the multiple existing paths. If there are many SLSs invoked, it might be the case that more resources are required because of the over-subscription ratio. Then DRsM and DRtM will try to find more resources, but always within the ND's guidelines. If this procedure is not successful there are two alternatives: either the invocation is not accepted, if the situation occurred before the admission request; or, re-dimensioning is invoked, if the problem happened after admission as a result of many ingress nodes receiving simultaneous admission requests.

Policy management influences almost all of the parts of the previous scenario. A more concrete example is the following. If there is an administrator's policy according to which 10% of the overall network resources should always be available to best effort traffic, then ND needs to take that policy in mind during the calculation of the configuration. In addition, DRsM needs to be aware of this policy so that it does not allow dynamic requests for additional resources corresponding to other CoS to reduce this percentage of resources for best effort traffic.

## 8 Summary and Future Work

We first proposed a template for Service Level Specifications (SLSs), followed by a functional architecture for supporting the QoS required by contracted SLSs, while trying to optimise use of network resources. The management plane aspects of our architecture include SLS subscription, traffic forecasting, network dimensioning and dynamic resource & route management. Most of these are policy-driven. The control plane aspects include SLS invocation and packet routing while data plane aspects include traffic conditioning and PHB-based forwarding. The management plane aspects of our architecture can be thought as a detailed decomposition of the BB concept in the context of an integrated management and control architecture that aims to support both qualitative and quantitative QoS-based services. Many of the functional blocks of our architectural model are also features of BBs, the main difference being that a BB is seen as driven purely by customer requests whereas, in our approach, TE functions are continually aiming at optimising the network configuration and its performance.

We plan to experiment with and demonstrate the system both on commercial network testbeds, based on Cisco routers, and on laboratory testbeds using Linux-based routers. We will also use a simulated testbed to validate and fine-tune the proposed algorithms and to be able to deal with large-scale networks, stress conditions, faults, etc. The system is being designed using a number of technologies for communications between the FBs: CORBA is being used for the majority of management plane interactions, with LDAP for accessing the PSS and the SLS and network repositories (not explicitly shown in Figure 2). The interfaces to the routers is based on SNMP, COPS-PR and command-line interfaces with an adaptation layer presenting a consistent interface to the management plane which is independent of whether the underlying router is commercial or experimental. The interface between the adaptation layer and the management plane FBs uses COPS-PR for configuration actions and it is currently a design issue whether SNMP or the accounting messages of COPS-PR will be used for monitoring and statistics gathering. RSVP is assumed for SLS invocations although alternative lightweight protocols are also under investigation. The negotiations for SLS subscription are based on XML/HTTP.

Finally, it should be stated that the proposed DiffServ-oriented management and control framework is based on similar validated work we have undertaken in the past on ATM [GEORG99]. As such, we

are fairly confident that the proposed architectural framework will result in a workable solution for end-to-end QoS in a DiffServ MPLS-based Internet.

## Acknowledgements

This work was undertaken in the Information Society Technologies (IST) TEQUILA project, which is partially funded by the Commission of the European Union. We would also like to thank the rest of our TEQUILA colleagues who have also contributed to the ideas presented here.

## References

- [AUKI00] P. Aukia, M. Kodialam, P.V.N. Koppol, T.V. Lakshman, H. Sarin, and B. Suter. "RATES: A Server for MPLS Traffic Engineering", IEEE Network Magazine, March/April 2000.
- [AWDU00] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, "A Framework for Internet Traffic Engineering", draft-ietf-tewg-framework-02.txt, Work in Progress, July 2000.
- [FELD00] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. "NetScope: Traffic Engineering for IP Networks", IEEE Network, March/April 2000.
- [FLEG01] P. Flegkas, P. Trimintzios, G. Pavlou, I. Andrikopoulos, C.F. Cavalcanti, "On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks", Proc. of the Workshop on Policies for Distributed Systems and Networks, Springer-Verlag (LNCS series), January 2001.
- [GEORG99] P. Georgatsos, D. Makris, D. Griffin, G. Pavlou, S. Sartzetakis, Y. T'Joens, D. Ranc, "Technology Interoperation in ATM Networks: the REFORM System", IEEE Communications, Vol. 37, No. 5, pp. 112-118, IEEE, May 1999
- [GODE00a] D. Goderis (ed.), "Functional Architecture and Top Level Design", TEQUILA Deliverable D1.1, September 2000, available at: [www.ist-tequila.org/deliverables.html](http://www.ist-tequila.org/deliverables.html)
- [GODE00b] D. Goderis, Y. T'Joens, C. Jacquenet, G. Memenios, G. Pavlou, R. Egan, D. Griffin, P. Georgatsos, L. Georgiadis, P. Van Heuven, "Service Level Specification Semantics and Parameters", draft-tequila-sls-00.txt, Work in Progress, November 2000
- [QBONE] B. Teitelbaum, "Qbone Architecture (v1.0)", 1999, available at: <http://www.internet2.edu/qos/wg/papers/qbArch/>
- [RFC-2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [RFC-2638] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, July 1999.
- [ROSE00] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", draft-ietf-mpls-arch-07.txt, Work in Progress, July 2000.