A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks

Panos Trimintzios, Ilias Andrikopoulos, George Pavlou, and Paris Flegkas, University of Surrey, U.K.
David Griffin, University College London, U.K.
Panos Georgatsos, Algonet S.A., Greece
Danny Goderis and Yves T'Joens, Alcatel, Belgium
Leonidas Georgiadis, Aristotle University of Thessaloniki, Greece
Christian Jacquenet, France Telecom R&D, France
Richard Egan, Thales Research, U.K.

ABSTRACT

As the Internet evolves toward the global multiservice network of the future, a key consideration is support for services with guaranteed quality of service. The proposed differentiated services framework is seen as the key technology to achieve this. DiffServ currently concentrates on control/data plane mechanisms to support QoS, but also recognizes the need for management plane aspects through the bandwidth broker. In this article 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 services while optimizing the use of underlying network resources.

INTRODUCTION

With the prospect of becoming the ubiquitous all-service network of the future, the Internet needs to evolve to support services with guaranteed quality of service (QoS) characteristics. The Internet Engineering Task Force (IETF) has proposed a number of QoS models and supporting technologies, including the integrated services (IntServ) and differentiated services (DiffServ) frameworks [1]. 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 behaviors (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 (SLSs) 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 DiffServ codepoint (DSCP) field of the IP header.

In order to achieve QoS guarantees, control plane mechanisms have been used to reserve resources on demand, but management plane mechanisms are also necessary to plan and provision the network, and manage requirements for service subscription according to available resources [2]. 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 telecommunications networks. Considering in particular the DiffServ architecture (Fig. 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 any architecture for end-to-end QoS delivery. In order to provide end-to-end quantitative QoS

0163-6804/01/$10.00 © 2001 IEEE

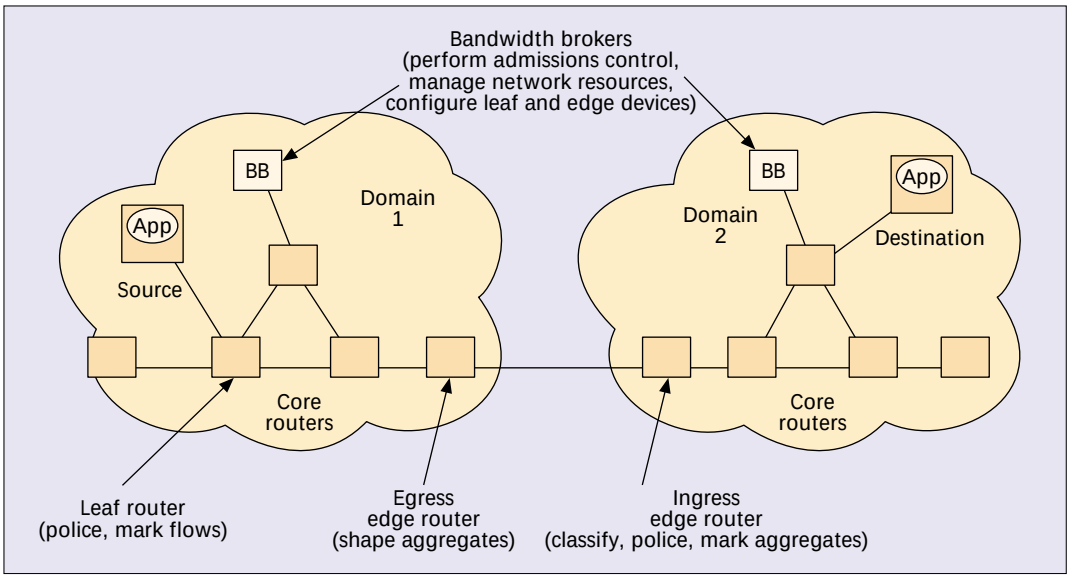

Figure 1. *The DiffServ architecture.*

The proposed framework ensures that agreed SLSs are adequately provisioned and that future SLSs may be negotiated and delivered through a combination of static, quasi-static and dynamic traffic engineering techniques both intra- and inter-domain.

guarantees, DiffServ mechanisms should be augmented with intelligent traffic engineering functions.

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 [3]. 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 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 cost-effectively. Hence, the target is to accommodate as many traffic requests as possible by optimally using the available network resources. Both objectives are difficult to realize in a multiservice network environment.

Multiprotocol label switching (MPLS) [4] 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, MPLS is a suitable mechanism 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 DiffServ and MPLS, IP traffic engineering has attracted a lot of attention in recent years (see [5–7] for some examples). The Traffic Engineering for Quality of Service in the Internet at Large Scale (TEQUILA) project[1] is one of the projects in this area. 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.

[1] *For more information visit the TEQUILA Web site: http://www.ist-tequila.org*

This article 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 QoS 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 to 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: fulfillment and assurance of the SLSs. The proposed framework ensures that agreed upon SLSs are adequately provisioned and that future SLSs may be negotiated and delivered through a combination of static, quasi-static, and dynamic TE 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*").

Service-Level Specifications

In this section we substantiate the notion of SLS [1]. The definition of SLSs is the first step toward 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 standardized set of SLS parameters. An 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 deployment of bandwidth broker

The service schedule indicates the start time and end time of the service (i.e., when the service is available). This might be expressed as a collection of the following parameters: time of day range, day of week range, and month of year range.

(BB) capabilities [8] require a standardized set of semantics for SLSs to be negotiated both between 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 and mean downtime 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.

CONTENTS AND SEMANTICS

The contents of an SLS [9] include the essential QoS-related parameters, including scope and flow identification, traffic conformance parameters, and service guarantees. More specifically, an SLS has the following fields: Scope, Flow Descriptor, Traffic Descriptor, 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 [1] 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, say, Ethernet, or for unnumbered links like in, for example, Point-to-Point Protocol (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 (1|1), (1|*N*), and (*N*|1), respectively.

The *flow descriptor* (*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. An 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 be either behavior aggregate (BA) or multifield (MF) based.

The *traffic descriptor* includes *traffic envelope* and *traffic conformance*, and 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* (described below). The TC parameters are the input to the *traffic conformance testing* algorithms. Traffic conformance testing is the set of actions which uniquely identifies the "in-profile" and "out-of profile"[2] (or excess) packets of an IP stream identified by the FlowDes. The TC parameters describe the reference values with which the traffic identified by the FlowDes will have to comply. The TC algorithm is the mechanism enabling unambiguous identification of all in- or out-of-profile packets based on these conformance parameters. The following is a nonexhaustive list of potential conformance parameters: *peak rate p* in bits per second, *token bucket rate r* (b/s), *bucket depth b* (bytes), *minimum MTU* — maximum transfer nnit — *m* (bytes), and *maximum MTU M* (bytes).

An *excess treatment* parameter describes how the service provider will process excess or out-of-profile traffic (or other than in-profile in the case of multilevel 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, such as the DSCP value in case of remarking 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*. Delay and jitter indicate the maximum packet transfer delay and packet transfer delay variation from ingress to egress, respectively. Delay and jitter may be specified as either worst-case (deterministic) bounds or quantiles. 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. For each of the four performance parameters a *time interval* can be also defined (Table 1).

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, the performance parameters delay and packet loss may be qualified. Possible qualitative values for delay and/or loss are *high*, *medium*, and *low*. The actual quantification of the relative difference between high, medium, and low is a policy-based decision (e.g., high = 2 × medium; medium = 3 × low). If the performance parameters are not quantified or qualified, the service will be best effort.

The *service schedule* indicates the start time and end time of the service (i.e., when the service is available). This might be expressed as a collection of the following parameters: time of day range, day of week range, and month of the year range. *Reliability* indicates the mean downtime (MDT) per year and the time to repair

[2] *Note that the conformance result might not necessarily be of a binary mode (in/out) but could also be multilevel (e.g., using a Two-Rate Three-Color Marker algorithm).*

	Virtual leased line service	Bandwidth pipe for data services	Minimum rate guaranteed service	Qualitative Olympic services		The funnel service
Comments	Example of a unidirectional 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 bulk ftp traffic, or adaptive video with min throughput requirements	They are meant to qualitatively differentiate between applications such as:		It is primarily a protection service; it restricts the amount of traffic entering a customer's network
				Online Web browsing	E-mail traffic	
Scope	(1|1)	(1|1)	(1|1)	(1|1) or (1|N)		(N|1) or (all|1)
Flow descriptor	EF, S-D IP-A	S-D IP-A	AF1x	MBI		AF1x
Traffic descriptor	(b, r) e.g. r=1	NA	(b, r)	(b, r), r indicates a minimum committed Olympic rate		(b, r)
Excess treatment	Dropping	NA	Remarking	Remarking		Dropping
Performance parameters	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
Service schedule	MBI, e.g., daily 9:00-17:00	MBI	MBI	MBI	MBI	MBI
Reliability	MBI, e.g., MDT = 2 days	MBI	MBI	MBI	MBI	MBI

(b, r): token bucket depth and rate (Mb/s), p: peak rate, D: delay (ms), L: loss probability, R: throughput (Mb/s), t: time interval (min), q: quantile, S-D: source and destination, IP-A: IP address, MBI: may be indicated, NA: not applicable, MDT: maximum down time (per year), ET: excess treatment, TC: traffic conformance

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

(TTR) in case of service breakdown. Other parameters might also be included in the SLS, such as the *assurance level*, which describes the percentage of the time the ISP will be able to conform to the other SLS parameters.

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 Fig. 2. There are three main parts in this architecture: SLS management (SLSM), 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 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 that 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 and then processed by the traffic forecast and 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.

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 interdomain 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 AS, or a business or residential user. The subscription (or registration) concerns the service-level agreement (SLA), containing 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 an 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.

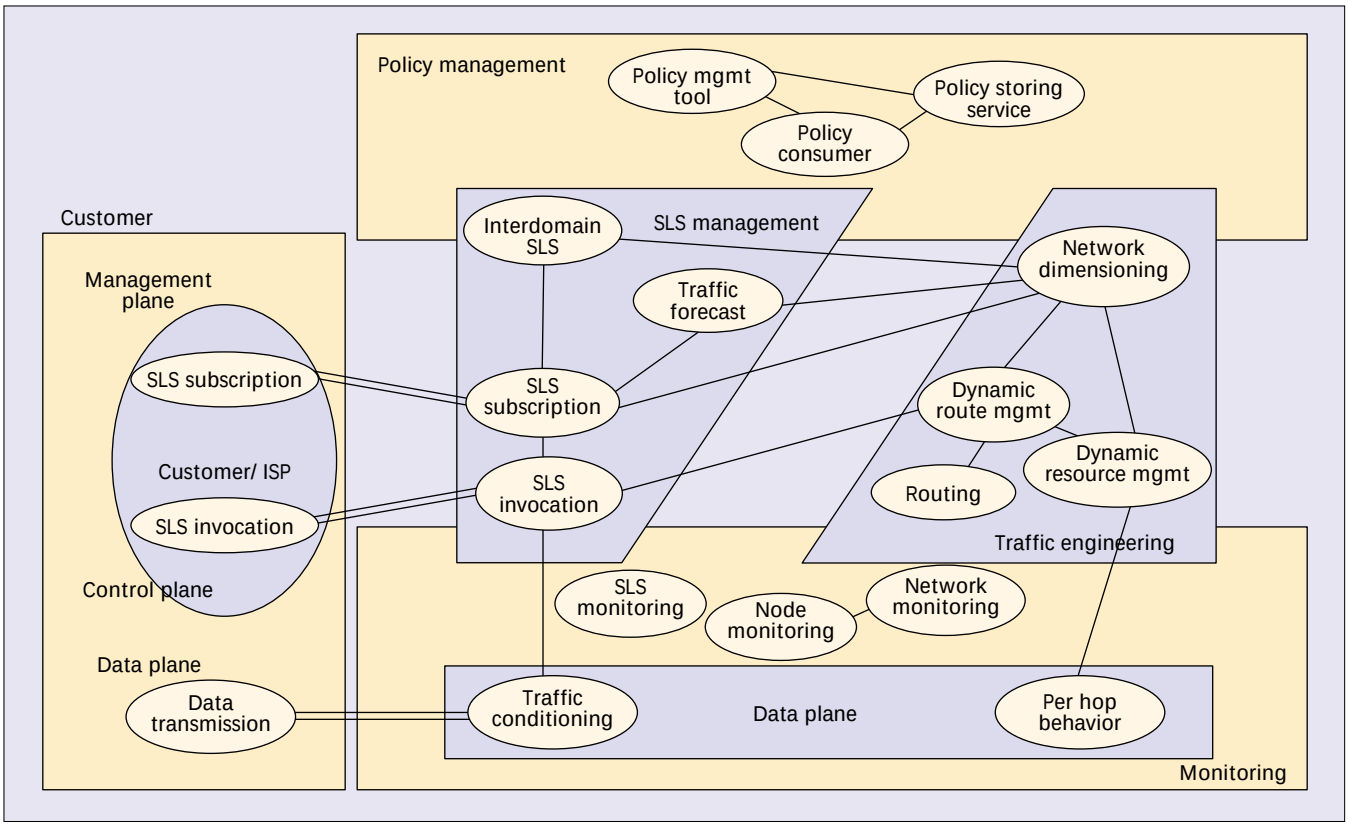

Figure 2. *The TEQUILA functional architecture.*

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 since it provides a guarantee that network resources will be available when required.

SLS invocation (SLS-I) is the FB that includes the process of dynamically dealing with a flow and is part of *control plane* functionality. It performs dynamic admission control as requested by the user; this process can be flow-based. SLS-I receives input from SLS-S (e.g., for authentication purposes) and has a view of 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 out-of-profile packets are handled in a certain way and so on. Both SLS-S and SLS-I interact with the *interdomain SLS requester*, which deals with all interdomain 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, physical nature, and capacities of the access links, business policies, economic models, and so on.

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

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

MPLS TE is exercised on two timescales, 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 offline taking into account global network conditions and traffic load. It involves 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 utilization and 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 TE. Triggered by the inability to adapt appropriately to significant changes in expected traffic load, or local changes in network topology, ERPs may be created or torn down by long-term TE functions.

Long-term TE corresponds to the *time-based* capacity management functions of TE [3], while short-term TE corresponds to *state-dependent* capacity management functions of TE. By virtue of our model, these functions interoperate toward a complete TE solution.

Network Dimensioning

Network dimensioning (ND) is responsible for mapping the traffic onto the physical network resources and provides network provisioning directives 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 centralized for a particular AS, although distributed implementations on a subdomain or area of an AS are also possible. In any case, it utilizes networkwide information, received from the network routers and/or other functional components through polling and/or asynchronous events.

ND is invoked in order of several hours (*short-term*) to days or weeks (*long-term*). Its main task is to accept input about the forecast demand from TF and, by knowing the physical topology, to calculate the configuration required by the elementary TE functions in a policy-driven fashion. The output of ND is the set of ERPs and their associated parameters in the form of directives. The objective of such calculation is to accommodate all the expected demand, and therefore meet the SLS performance requirements, without overloading any part of the network. Providing directives and not specific "hard" values leaves space for unpredictable traffic fluctuations, handled by dynamic route and resource management (DRtM, DRsM), and at the same time not having to reroute large amounts of traffic in the case of failures. One can formulate the dimensioning problem as an optimization problem and solve it by using either optimization techniques or heuristic algorithms to overcome any complexity problems. The definition, analysis, and testing of such algorithms and techniques is part of the ongoing work in the context of the TEQUILA project.

The output of ND is fed to DRtM and DRsM to handle dynamic changes, 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.

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 with 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, of overutilization 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.

Dynamic Resource Management

DRsM has distributed functionality, with an instance attached to each router. This component aims to ensure 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

Network dimensioning is responsible for mapping the traffic onto the physical network resources and provides network provisioning directives in order to accommodate the forecasted traffic demands.

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.

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 timescale. 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 an 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, the allocated buffer space may need to change.

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.

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, such as SLS-related admission policies for SLS management, dimensioning policies for ND, dynamic resource/route management policies for DRsM/DRtM, and so on.

Although Fig. 2 has shown a single PC/PDP for illustrative purposes, our model assumes many instances of policy consumers [10]. In reality, the PC/PDP is not a separate component but is collocated with other functional blocks (e.g., SLS-S and SLS-I, TF, ND, DRtM, and DRsM). Targets can be the managed objects of the associated FB or 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 also be 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 runtime. 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 [10]. 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 Fig. 2.

A WORKING SYSTEM SCENARIO

In this section we 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 redimensioning needs to be done. The reasons for redimensioning might be:

- The amount of spare resources for future SLS subscriptions is below a (policy-based) defined threshold.
- The amount of SLS subscription rejections is greater than a (policy-based) defined threshold.
- DRtM or DRsM is unable to handle the current resource demand.
- The redimensioning cycle has elapsed (dimensioning period).

First, ND will request the traffic forecast that corresponds to the next dimensioning period. TF will consider the currently subscribed N SLSs, the (policy-based) additional M SLS subscription requests predicted for the next dimensioning period, the (policy-based) oversubscription ratio, and historical monitoring data in order to prepare the traffic forecast matrices (one per CoS). The demand provided to ND will be something between (N, $N + M$). ND will use some optimization or heuristic dimensioning algorithm in order to define multiple paths (trees) between the ingress and (list of) egress nodes as well as the estimate of required resources for each PHB at each node (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 in the form of directives, giving the space to operate, which contact the network elements (NEs) in order to enforce these directives by setting up 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 oversubscription ratio. Then DRsM and DRtM will try to find more resources, but always within the ND's guidelines and directives. 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, redimensioning (most probably short-term or long-term if the problem is more severe) 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 percent of overall network resources should always be available to best effort traffic, ND needs to keep that policy in mind during 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 CoSs to reduce this percentage of resources for best effort traffic.

Summary and Future Work

In this article we propose a template for service-level specifications, followed by a functional architecture for supporting the QoS required by contracted SLSs, while trying to optimize use of network resources. The management plane aspects of our architecture include SLS subscription, traffic forecasting, network dimensioning, and dynamic resource and route management. All 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 of 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 continually aim at optimizing the network configuration and its performance.

We plan to experiment with and demonstrate the system on both commercial network testbeds, based on Cisco routers, and 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, and so on. The system is being designed using a number of technologies for communications between the FBs. Common Object Request Broker Architecture (CORBA) is being used for the majority of management plane interactions, with Lightweight Directory Access Protocol (LDAP) for accessing the PSS and SLS and network repositories (not explicitly shown in Fig. 2). The interfaces to the routers are based on the Simple Network Management Protocol (SNMP), Common Open Policy Service Protocol for Provisioning (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 a current design issue is 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 the Extensible Markup Language (XML).

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 [2]. 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.

We plan to experiment with and demonstrate the system on both commercial network testbeds, based on Cisco routers, and laboratory testbeds using Linux-based routers. We will also use a simulated testbed to validate and fine-tune the proposed algorithms.

Acknowledgments

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

[1] S. Blake *et al.*, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.
[2] P. Georgatsos *et al.*, "Technology Interoperation in ATM Networks: the REFORM System," *IEEE Commun. Mag.*, vol. 37, no. 5, May 1999, pp. 112–18.
[3] D. Awduche *et al.*, "A Framework for Internet Traffic Engineering," draft-ietf-tewg-framework-02.txt, work in progress, July 2000.
[4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.
[5] P. Aukia *et al.*, "RATES: A Server for MPLS Traffic Engineering," *IEEE Network*, Mar./Apr. 2000.
[6] A. Feldmann *et al.*, "NetScope: Traffic Engineering for IP Networks," *IEEE Network*, Mar./Apr. 2000.

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.

[7] B. Teitelbaum, "Qbone Architecture (v1.0)," 1999; http://www.internet2.edu/qos/wg/papers/qbArch.

[8] K. Nichols, V. Jacobson, and L. Zhang, "A Two-Bit Differentiated Services Architecture for the Internet," RFC 2638, July 1999.

[9] D. Goderis *et al.*, "Service Level Specification Semantics and Parameters," draft-tequila-sls-00.txt, work in progress, Nov. 2000.

[10] P. Flegkas *et al.*, "On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks," *Proc. Wksp. Policies for Dist. Sys. and Networks*, Springer-Verlag LNCS series, Jan. 2001.

BIOGRAPHIES

PANOS TRIMINTZIOS (P.Trimintzios@eim.surrey.ac.uk) received a B.S in computer science and an MSc in computer networks from the University of Crete, Greece, in 1996 and 1998, respectively. From 1995 to 1998 he was a research associate at ICS-FORTH, Greece, working on projects involving high-speed network management and charging network and user services. Currently he is a research fellow at the Centre for Communication Systems Research (CCSR), University of Surrey, United Kingdom, where he is also a Ph.D. candidate. His main research interests include IP traffic engineering, constraint-based routing, IP QoS provisioning, network performance control and management, and service offering and negotiation.

ILIAS ANDRIKOPOULOS (ilias@ieee.org) holds a diploma in physics from the University of Athens, Greece, an M.S. in information technology from University College London, and a Ph.D. in electronic engineering with specialization in networking from the University of Surrey. While studying for his Ph.D., he was a research fellow in the Networks Research Group at CCSR, University of Surrey, working in EU and U.K.-funded research projects. His main research interests include IP quality of service, traffic management, Internet technologies, mobile and satellite networking, and network management.

GEORGE PAVLOU (G.Pavlou@eim.surrey.ac.uk) is professor of communication and information systems at the CCSR, School of Electronics and Computing, University of Surrey, where he leads the activities of the Networks Research Group. He received a diploma in electrical engineering from the National Technical University of Athens, Greece, and M.S. and Ph.D. degrees in computer science from University College London. His research interests include network planning and dimensioning, traffic engineering and management, programmable and active networking, multimedia service control, and technologies for object-oriented distributed systems. He has contributed to standardization activities in ISO, ITU-T, TMF, OMG, and IETF and is technical program co-chair of IEEE/IFIP Integrated Management 2001.

PARIS FLEGKAS (P.Flegkas@eim.surrey.ac.uk) received a diploma in electrical and computer engineering from Aristotle University, Thessaloniki, Greece, and an M.S. in telematics (communications and software) from the University of Surrey in 1998 and 1999, respectively. He is currently a Ph.D. student at the University of Surrey, and his research interests are in the areas of policy-based management, traffic engineering, and IP QoS.

DAVID GRIFFIN (D.Griffin@ee.ucl.ac.uk) received his B.S. degree in electrical engineering from Loughborough University, United Kingdom, in 1988. Over the past 12 years his research — initially at Marconi Communications, United KIngdom, and then at ICS-FORTH, Greece — has covered TMN and TINA systems, ATM performance management, IP traffic engineering, distributed processing, and mobile code. He joined University College London in 1996 and is currently a senior research rellow in the Department of Electronic and Electrical Engineering, where he is also completing a Ph.D. His main research interest is in the planning, management, and dynamic control for providing QoS in multiservice networks.

PANOS GEORGATSOS (pgeorgat@algo.com.gr) received a B.S. degree in mathematics from the National University of Athens in 1985, and a Ph.D. in computer science from Bradford University, United Kingdom, in 1989. He is currently working at Algonet S. A., Athens, Greece, where he is responsible for the R&D Group in Telecommunications. His research interests include service quality management, network routing, planning, resource dimensioning, analytical modeling, simulation, and architectures for distributed systems.

DANNY GODERIS (Danny.Goderis@alcatel.be) received an M.S. degree in physics and a Ph.D. in mathematical physics from the Catholic University Leuven, Belgium, in 1986 and 1990. After some more years at the University he joined the Alcatel Corporate Research Centre in 1998. He is currently responsible for IP QoS technologies in the Alcatel Corporate Network Strategy Group and is leading the IP traffic performance studies. He is project manager of the European IST-project TEQUILA.

YVES T'JOENS (Yves.TJoens@alcatel.be) holds a B.S. in aeronautical engineering from the Universities of Gent, Brussels and Leuven, Belgium in 1992, and an M.S. in technology from the University of Manchester in 1993. He is with Alcatel Bell, Antwerp, Belgium. From 1994 to 1996 he worked in the Broadband Switching Division on signaling, call handling, and routing for ATM. In 1996 he joined the corporate research center, taking up standardization of ATM routing, signaling, and survivability in the ATM Forum and ITU-T. In 1998 he was appointed to the position of Internet Access leader for research on advanced Internet access architectures. He has contributed to standardization in the ATM Forum, ADSL Forum, and IETF.

LEONIDAS GEORGIADIS [M'86, SM'95] (leonid@eng.auth.gr) received his degree in electrical engineering from Aristotle University, Thessaloniki, Greece, in 1979, and his M.S. and Ph.D. degrees, both in electrical engineering, from the University of Connecticut, in 1981 and 1986, respectively. From 1986 to 1987 he was research assistant professor at the University of Virginia, Charlottesville. In 1987 he joined IBM T. J. Watson Research Center, Yorktown Heights, New York, as a research staff member. Since October 1995 he has been with the Telecommunications Department of Aristotle University, Thessaloniki, Greece. His interests are in the areas of high-speed network management, scheduling, congestion control, mobile communications, modeling, and performance analysis.

CHRISTIAN JACQUENET (Christian.Jacquenet@francetelecom.fr) graduated in 1987 from the Ecole Nationale Supérieure de Physique de Marseille. In 1989, he joined France Telecom where he was in charge of the specification and deployment of technical support related to the IP internetworking service offering of France Telecom. In 1993 he joined the research labs of France Telecom (FTR&D) and, from 1993 to 1997, he was involved in the specification and evaluation of ATM-based internetworking service offerings. He is currently the head of an R&D team, which is in charge of the specification development and validation of value-added IP service offerings, including QoS-based IP VPN, label-based switching techniques, IPv6- and multicast-based networks, as well as dynamic provisioning techniques for the enforcement of policies related to traffic engineering and QoS.

RICHARD EGAN (Richard.Egan@rrl.co.uk) received a B.eng. (Elect.) from University College, Cork, Ireland in 1980. He has worked for GEC Telecommunications and Racal Datacom on a variety of product developments. Since joining Thales Research (formerly Racal Research Ltd.) in 1992, he has led a team that specializes in system design and performance analysis of data networks. His main interests are in QoS, VoIP, traffic engineering, and SLA development.