The Structure and Management of Service Level Agreements in Networks

Eric Bouillet   Debasis Mitra   K. G. Ramakrishnan

Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974-0636 USA
mitra@research.bell-labs.com

Abstract

The paper proposes a structure for QoS-centered Service Level Agreements (SLA), and a framework for their real time management in multiservice packet networks. The SLA is structured to be fair to both parties, the service provider and their customer. The SLA considered here are for QoS assured delivery of aggregate bandwidth from ingress to egress nodes; however, the control and signalling is for the more granular flows or calls. A SLA monitoring scheme is presented in which revenue is generated by the admission of flows into the network, and penalty incurred when flows are lost in periods when the service provider is not SLA compliant. In the SLA management scheme proposed here the results of a prior off-line design are used, in conjunction with measurements taken locally at ingress nodes, to classify the loading status of routes. The routing and resource management are based on Virtual Partitioning and its supporting mechanism of Bandwidth Protection. The effectiveness of SLA management is measured by the robustness in performance in the presence of substantial diversity in actual traffic conditions. A simulation testbed called D'ARTAGNAN has been built from which we report numerical results for a case study. The results show that the SLA management scheme is robust, fair and efficient over a broad range of traffic conditions.

Keywords. SLA, QoS, VPN, network design, Virtual Partitioning, Bandwidth Protection

1 Introduction

This paper presents a framework for constructing and managing QoS-centered Service Level Agreements (SLA) between service providers, who own and operate the network infrastructure, and their customers. The main elements of the framework are (see also Figure 1): (i) We propose a structure for SLA, which balances the interests of service providers and their customers. It gives specifications for the QoS-assured transfer of the customer's traffic in each of several QoS classes and for each of many, possibly all, network ingress-egress node pairs. (ii) The twin drivers for the SLA management scheme are, first, revenue, i.e., payments from the customer to the service provider, which is generated by carrying traffic

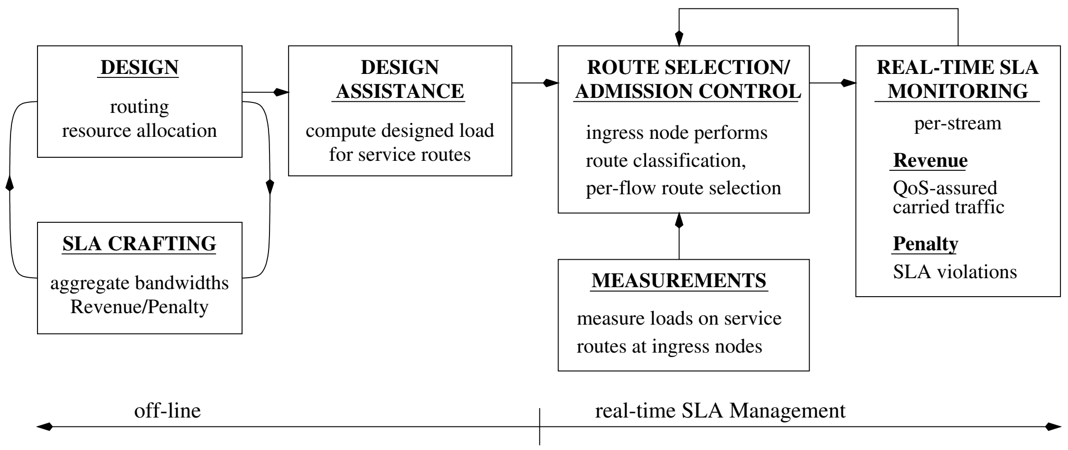

Figure 1: Elements in Design Assisted, Real Time SLA Management

regardless of whether the offered amount is more or less than is stipulated in the SLA, and, second, penalties, i.e., payments from the service provider to the customer, from SLA violations. Both measures are computed at the granularity of a QoS class and node pair; the net revenue is the overall performance measure. (iii) Before the SLA is initiated, it is assumed that a process of due diligence involving design has taken place, as a consequence of which it is assured that the service provider has resources, which, when combined with preplanned routing, will allow SLA-stipulated aggregate traffic to be carried at a mutually satisfactory level of QoS and net revenue. (iv) We describe the process by which the results of the aforementioned off-line design process is used in the real time SLA management and also in crafting the SLA. (v) The selection of routes for flows is source-based, and it requires the ingress node to measure locally the load imposed by it on each route that originates with it, and then to compare the measured load to the designed load, and, finally, to determine the loading status of the route to be either *undersubscribed* or *oversubscribed.* (vi) A fundamental tenet is that the actual traffic offered to the network may be significantly different from SLA envisaged conditions, and that it is therefore in the interest of the service provider to deviate, possibly significantly if necessary, from resource allocations tacit in the original design. Our resource management scheme for the service provider is based on Virtual Partitioning and its supporting mechanism of Bandwidth Protection, which has the goal of carrying as much of the currently offered flows as is judicious given the possibility of incurring penalties in the future. The measure of effectiveness of SLA management is robustness in performance, as indicated by the net revenue, in the presence of great diversity in traffic conditions. (vii) Measurements have two main roles, first in classifying routes by loading status at network ingress, and, second, in SLA monitoring. We use both windows and exponential smoothing to determine the time scales in the respective averaging processes. (viii) We have built a comprehensive simulation testbed in software, called D'ARTAGNAN, for algorithm development and experimentation, from which we report extensively for a case study.

The present work is empirical with the primary focus being the development of a framework for SLA and new management algorithms that additionally couple and exploit extensive prior work, specially in design optimization [1, 2, 3, 4, 5, 6, 7] and resource management [8, 9, 10, 11, 12]. Our empirical results show that the scheme for SLA management is robust, fair and efficient over a broad range of traffic conditions.

Network controls, i.e. route selection and admission control, are exercised at the granularity of what we term flows or calls, while the SLA is for aggregate or bulk bandwidth, which is typically several orders of magnitude coarser than flows. The definition and size of flows in our framework may be adapted to scalability requirements. When flows are defined finely (see [13] for examples), QoS is finely controlled. On the other hand, when flows are coarsely sized, the burden on network control and signalling is relaxed. In any case, in the framework of this paper flows have random arrivals and lifetimes or holding times. Flows belong to various application classes, each with characteristic QoS requirements, as well as mean arrival rates and holding times. The notion of Effective Bandwidth is used to encapsulate various packet level issues, such as burstiness and QoS (delay, jitter, loss) at network elements [14, 15]. We assume that each flow of service type s requires (effective) bandwidth d_s. QoS at the flow level is obviously also important, and this is handled in the SLA, as described later, by the flow acceptance ratio. The main steps in handling a new flow are (i) route selection, and (ii) admission control/bandwidth reservation in all links in the route during flow set-up [16], which requires signalling. We give a measurement-based procedure for route selection that is implemented at ingress nodes and favors undersubscribed routes and previously successful routes ("sticky routing"). Admission control is implemented by links exercising Bandwidth Protection in which flows on undersubscribed routes are unconditionally accepted, while for oversubscribed routes acceptance is conditional on their being spare bandwidth sufficient to carry R future flows of any QoS class, where R is a parameter having value of about unity. This mechanism is at the core of our adaptation of Virtual Partitioning [9, 10, 11] for real time SLA management.

One of our major empirical results is that in a variety of traffic loading scenarios characterized by significant departures in offered traffic from the nominal SLA-stipulated values, robustness is achieved, i.e. revenues are high and penalties small, when the bandwidth protection parameter R is 1 or 2.

Of course, Virtual Private Networks [5] and SLA go hand in hand. Our point of view, however, is that it is more useful to consider first the present case where, in effect, there is a SLA for each stream, which (see below) is the composite of QoS class and ingress-egress node pair. Multiple VPN, each having its own SLA with the service provider, may be handled simply by extending the definition of a stream given here to include additionally the VPN index. Admittedly there are nuances not treated here, such as differentiated levels of resource sharing between QoS classes and VPN, as postulated, for instance, in Hierarchical Virtual Partitioning [11], which await a future study.

Policy, QoS and other considerations constrain the routing of flows [17, 18, 19]. For instance, converged data networks will support Internet telephony, as well as packetized video and other delay sensitive services, in addition to delay insensitive data in a variety of grades of service. To handle such end-to-end constraints we rely on the notion of *admissible route sets*, $\mathcal{R}(s, \sigma)$, which are specific to each QoS service class s, and (ingress, egress) pair (σ_1, σ_2), which we denote by σ [3, 5, 6]. We refer to (s, σ) as a *stream* and (s, r), $r \in \mathcal{R}(s, \sigma)$, as a *service route*.

Section 2 describes the structure of the proposed SLA and the SLA monitoring process in which revenue and penalty are calculated. Section 3 describes the upstream processes of Design and SLA Crafting (see Fig. 1) from which the SLA management inherits useful information. Section 4 describes the procedure of route classification at ingress nodes based on measurements and assistance from the Design. Section 5 describes the key routing and admission control algorithms. Section 6 describes a case study and numerical results for it from D'ARTAGNAN. Section 7 gives concluding remarks.

2 Service Level Agreements: Revenue and Penalty

This section gives details of the SLA and its revenue and penalty clauses. The SLA stipulates two important parameters for each stream (s, σ), namely the aggregate (or bulk) offered and carried bandwidths, $U_{s\sigma}$ and $V_{s\sigma}$, $(V_{s\sigma} < U_{s\sigma})$, respectively. Their ratio $V_{s\sigma}/U_{s\sigma}$ is implicitly the contracted flow acceptance ratio for stream (s, σ). Tacit to SLA crafting must be the appreciation of the fact that as this quantity approaches unity, the required network resources approach infinity, a consequence of the uncertainty associated with the demand.

The basic idea in the proposed SLA is the following. If the customer of stream (s, σ) is requesting bandwidth not in excess of the contracted amount $U_{s\sigma}$, then the SP is expected to carry bandwidth that is in the same proportion to what is being requested as $V_{s\sigma}$ is to $U_{s\sigma}$. If, on the other hand, the customer is requesting stream bandwidth which is in excess of the contracted amount $U_{s\sigma}$, then the SP is only obligated to carry $V_{s\sigma}$.

To implement the above idea and to quantify compliance we rely on measurements, which are made over time periods called windows, and estimates are updated at the end of windows. Let $\tilde{U}_{s\sigma}(n)$ and $\tilde{V}_{s\sigma}(n)$ denote the estimates, based on measurements at the ingress node σ_1, of the offered and carried bandwidth, respectively, on stream (s, σ) at the end of the n^{th} SLA window. The details of the measurement procedure and the averaging implicit in it are given in Section 4.1. It is worth mentioning here that this procedure has two parameters, τ_{SLA} and α_{SLA}, for the window length and exponential smoothing, and both should be stipulated in the SLA since they jointly determine the time scale of the averaging in the SLA monitoring and accounting process.

We next introduce the notion of the $\text{SLA\_state}_{s\sigma}(n)$:

$$\text{SLA\_state}_{s,\sigma}(n) \triangleq (\text{customer\_state}_{s,\sigma}(n),\ \text{SP\_state}_{s,\sigma}(n))\,, \tag{1}$$

where each of the two state variables are binary valued, either compliant or non-compliant. For instance, if the measurements indicate that the customer of stream (s, σ) has requested in the n^{th} window total bandwidth not in excess of the SLA-stipulated amount of $U_{s\sigma}$, then the $\text{customer\_state}_{s,\sigma}(n) = \text{compliant}$, and otherwise it is 'non-compliant'. Table 1 describes the complete procedure for determining the state variables based on the measured quantities $(\tilde{U}_{s\sigma}(n),\ \tilde{V}_{s\sigma}(n))$ and the SLA parameters $(U_{s\sigma},\ V_{s\sigma})$.

Table 1: SLA_STATES

		SP_state (n)	
		COMPLIANT	NON-COMPLIANT
Customer_state(n)	COMPLIANT	$\tilde{U}_{s\sigma}(n) \leq U_{s\sigma}$ $V_{s\sigma}/U_{s\sigma} \leq \tilde{V}_{s\sigma}(n)/\tilde{U}_{s\sigma}(n)$	$\tilde{U}_{s\sigma}(n) \leq U_{s\sigma}$ $V_{s\sigma}/U_{s\sigma} > \tilde{V}_{s\sigma}(n)/\tilde{U}_{s\sigma}(n)$
	NON-COMPLIANT	$\tilde{U}_{s\sigma}(n) > U_{s\sigma}$ $V_{s\sigma} \leq \tilde{V}_{s\sigma}(n)$	$\tilde{U}_{s\sigma}(n) > U_{s\sigma}$ $V_{s\sigma} > \tilde{V}_{s\sigma}(n)$

In our framework the SLA_state of the stream is uniform over each window. This is so for scalability reasons, i.e., for controlling the processing burden, and also to allow averaging and greater accuracy. Similarly, the revenue and penalty measures are updated only at the end of every window, as is the SLA_state.

Regardless of the SLA_state, every flow that is carried generates revenue and increments the revenue measure. This is important, since it is responsible for a persistent incentive to carry all offered flows, and it has the effect of encouraging resource sharing. Each flow that is carried on the stream (s, σ) generates revenue $w_{s\sigma}$. This service sensitive quantity has a default value, which is the product of d_s, the effective bandwidth of calls of service s, and h_s, the mean holding time for such calls. Each increment of the penalty is $m_{s\sigma} w_{s\sigma}$, where $m_{s\sigma}$ is the important *penalty multiplier*, typically exceeding unity. This parameter is also specified in the SLA and is important in tuning the SLA management techniques. When the penalty multipliers are unity, there is no incentive for the SLA management to be sensitive to the particulars of the SLA, while, at the other extreme, when the penalty multipliers are large, the SLA management and network controls need to be very careful about committing resources in excess of the values obtained in the design and thereby causing the SP_state to turn non-compliant and flows to be lost at some point in the future.

We envisage that the ingress node σ_1 of each stream (s, σ) maintains a cumulative count of the revenue $W_{s\sigma}(n)$ and the penalty$_{s\sigma}(n)$, which are updated at the end of the n^{th} window.

The cumulative measures for revenue and penalty are calculated as follows:

$$W_{s\sigma}(n) - W_{s\sigma}(n-1) = w_{s\sigma} \tau \tilde{V}_{s\sigma}(n) \tag{2}$$

and

$$
\text{penalty}_{s\sigma}(n) - \text{penalty}_{s\sigma}(n-1) =
\begin{cases}
0 \quad \text{if} \quad \text{SP\_state}_{s,\sigma}(n) = \text{compliant} \\
m_s w_{s\sigma} \tau \left[\left(\frac{V_{s\sigma}}{U_{s\sigma}} \right) \tilde{U}_{s\sigma}(n) - \tilde{V}_{s\sigma}(n) \right] \\
\qquad\qquad \text{if SLA\_state}_{s\sigma}(n) = (\text{compliant, non-compliant}) \\
m_s w_{s\sigma} \tau \left[V_{s\sigma} - \tilde{V}_{s\sigma}(n) \right] \quad \text{if SLA\_state}_{s,\sigma}(n) = (\text{non-compliant, non-compliant})
\end{cases}
\tag{3}
$$

The important measure for the stream (s, σ) is the net revenue,

$$
W\_\text{net}_{s\sigma}(n) = W_{s\sigma}(n) - \text{penalty}_{s\sigma}(n)\,. \tag{4}
$$

Summing over all services s and ingress–egress node pairs σ, gives the important network-wide measures $W(n)$, penalty(n) and $W\_\text{net}(n)$.

In concluding this section we note that there are important issues related to pricing of SLA that are outside the scope of this work. For instance, it is in the interest of the customer to have high values for the parameters $(U_{s\sigma}, V_{s\sigma})$, which however burden the SP. Hence, we expect that the SLA is accompanied by payments other than the revenue and penalty payments that have been the subject of this section.

3 Design

3.1 Model and Design Problem

The coupled processes of Design and SLA Crafting (see Figure 1) are important, but outside the primary focus of this paper. Yet these topics cannot be entirely avoided since they share a common framework with SLA management, the primary focus here, and, specifically, the design is used in an essential way here. We deal with this by describing the design problem, as well as the results from it that we assume are transferred to the SLA management process, while leaving open the selection of techniques. We mention that the reader will find in [3] the design techniques for a stochastic framework, and in [6] simpler, faster, albeit less accurate techniques based on multicommodity flow models.

In the design problem formulation it is assumed that requests for calls for stream (s, σ) arrive at the ingress σ_1 as a Poisson process with rate $\Lambda_{s\sigma}$. Each call requires bandwidth d_s and has mean holding time h_s. The aggregate bandwidth requested for the stream (s, σ)

is thus (see also Section 2)

$$U_{s\sigma} = \bar{\rho}_{s\sigma} d_s \tag{5}$$

where we have defined the stream call intensity $\bar{\rho}_{s\sigma} \triangleq \Lambda_{s\sigma} h_s$. The design objective is to allocate the offered traffic to routes r in the admissible set $\mathcal{R}(s,\sigma)$, i.e., to determine λ_{sr}, the rate of calls to the service route (s,r) such that

$$\sum_{r \in \mathcal{R}(s,\sigma)} \lambda_{sr} \leq \Lambda_{s\sigma}, \tag{6}$$

or, equivalently, $\sum \rho_{sr} \leq \bar{\rho}_{s\sigma}$, where ρ_{sr} is the call intensity on (s,r), $\rho_{sr} \triangleq \lambda_{sr} h_s$.

Congestion effects in the network are responsible for the traffic offered to the service route (s,r) to be thinned by the factor $(1 - L_{sr})$ to give the actual carried traffic on (s,r). Here L_{sr} is the loss ratio (or probability) on service route (s,r) and, of course, depends on all the traffic in the network. Let M_{sr} denote the aggregate bandwidth carried on (s,r) that the design process yields,

$$M_{sr} = \rho_{sr}(1 - L_{sr}) d_s \tag{7}$$

Similarly, the carried aggregate bandwidth on stream (s,σ) that the design process obtains is denoted by $M_{s\sigma}$, where

$$M_{s\sigma} = \sum_{r \in \mathcal{R}(s,\sigma)} M_{sr} \tag{8}$$

On comparing (5) and (8) we see that for stream (s,σ) the ratio (or probability) of acceptance in the design is $M_{s\sigma}/U_{s\sigma}$.

An objective that may be used to optimize the design, as in [3] and [6], is the maximization of the long term average rate of network-wide revenue generation, wherein each flow of stream (s,σ) that is carried generates revenue $w_{s\sigma}$. This quantity is also used in Section 2. Note that we have not directly incorporated the penalty generation process in the design stage, although this is certainly possible (see [9, 10, 11] for some directions), but at substantial additional complexity.

3.2 Design – SLA Interface

Note that the carried aggregate bandwidth measures, M_{sr} and $M_{s\sigma}$ in (7) and (8), are mean values. In corporating the results of the design in the SLA, attention must be paid to the variability in these measures. The design process should yield associated estimates of standard deviation, which may be used. In our experience [6] it turns out to be quite adequate to use instead $\sqrt{M_{sr}}$ and $\sqrt{M_{s\sigma}}$ as measures of variability.

We will need the key quantity X_{sr}, the designed load on service route (s, r), which is derived from the design thus [6]:

$$X_{sr} = M_{sr} + \gamma\sqrt{M_{sr}} \tag{9}$$

where γ is a small nonnegative number for which a typical value is 0.5. A similarly important quantity, which is stipulated in the SLA, is $V_{s\sigma}$, the contracted carried aggregate bandwidth for stream (s, σ). It is given by

$$V_{s\sigma} = M_{s\sigma} - \beta\sqrt{M_{s\sigma}} \tag{10}$$

where β is another small nonnegative number. Clearly increasing β is appropriate for reflecting increased aversion on the part of the SP to incur penalties, given the role of $V_{s\sigma}$ in the SLA monitoring process (see Section 2); on the other hand, increasing β negatively impacts the flow acceptance ratio $V_{s\sigma}/U_{s\sigma}$ in the SLA. We have found $\beta \approx 0.5$ to be typically appropriate.

In summary, the information passed from the design – SLA crafting process to the SLA management process are $\{U_{s\sigma}, V_{s\sigma}\}$, and $\{X_{sr}\}$. The latter is used, as we shall see in the next section, in assisting the network routing and admission control.

4 Route Classification at Ingress by Measurements and Design Assistance

Route classification denotes the process in which the ingress node, say σ_1, determines the (aggregate bandwidth load) status of every service route (s, r) that originates with it for every time window n. That is, σ_1 maintains a database of variables $\{\text{status}_{sr}(n)\}$, which is computed at the beginning of the n^{th} window and remains unchanged during the window, for all admissible routes $r \in \mathcal{R}(s, \sigma)$, for all egress nodes σ_2 and service class s. The status of the service route (s, r) is computed by comparing X_{sr}, its designed bandwidth load, to the measured load $Z_{sr}(n)$. The computation of X_{sr} has been described in the preceding section, and later in this section the procedure for computing $Z_{sr}(n)$ is described.

Specifically, $\text{status}_{sr}(n)$ is binary valued,

$$\text{status}_{sr}(n) \in \{\text{undersubscribed (US), oversubscribed (OS)}\} \tag{11}$$

and,

$$\begin{aligned} \text{status}(n) &= \text{US} \quad \text{if} \quad Z_{sr}(n) \leq X_{sr} \\ &= \text{OS} \quad \text{if} \quad Z_{sr}(n) > X_{sr} \end{aligned} \tag{12}$$

This variable is central to the routing and admission control functions performed at the ingress, as described in the next section. It is also noteworthy that the computation of $Z_{sr}(n)$ is based on averaging over the window only local measurements at the ingress node of the bandwidth requirements of currently active calls on the service route (s, r).

4.1 Measurements at the Ingress

We briefly describe a measurement procedure based on windows (length τ) and exponential smoothing (α). The description is for $Z_{sr}(n)$, although a similar process with possibly different parameters are used to compute $\{\tilde{U}_{s\sigma}(n), \tilde{V}_{s\sigma}(n)\}$, see Section 2, from measurements. All these quantities are computed from local measurements at the network ingress σ_1.

For any time t in the n^{th} window, i.e., $(n-1)\tau \leq t < n\tau$, we let $Y_{sr}(t)$ denote the aggregate bandwidth usage on service route (s, r) at t. This is a jump process which increments by d_s with each new flow and decrements by d_s with the departure of each flow. We let $\bar{Y}_{sr}(n)$ denote the mean bandwidth usage on (s, r) over the n^{th} window, i.e.,

$$\bar{Y}_{sr}(n) = \frac{1}{\tau} \int_{(n-1)\tau}^{n\tau} Y_{sr}(\eta) d\eta \tag{13}$$

Finally, we let $Z_{sr}(n+1)$ denote the exponentially smoothed estimate of aggregate bandwidth usage on (s, r) at the start of the $(n+1)^{\text{th}}$ window:

$$Z_{sr}(n+1) = \alpha Z_{sr}(n) + (1-\alpha)\bar{Y}_{sr}(n) \tag{14}$$

The influence of (α, τ) on the mean and variance of the above estimates for canonical stationary processes is a subject of great interest. Our results on this topic are outside the scope of this paper. See Benes [20] for some classical results.

The computation of the variables $\{\tilde{U}_{s\sigma}(n), \tilde{V}_{s\sigma}(n)\}$ used in monitoring SLA compliance is similar, although the window length and smoothing parameter, τ_{SLA} and α_{SLA}, are typically different.

5 Routing and Admission Control

Below we first review basic notions underlying the control algorithms before describing the latter.

5.1 Virtual Partitioning and Bandwidth Protection

Virtual Partitioning (VP), together with its supporting mechanism of Bandwidth Protection (BP), form the conceptual bases for our proposal for real time SLA management. VP is a resource-sharing concept, and in applying it here we view each link ℓ's bandwidth to be a resource, which is contended for by all service routes (s, r) whose path includes ℓ. Now, as described in the preceding section, the Design–SLA crafting process allocates bandwidth loads X_{sr} to these service routes, and, furthermore, the comparison of these loads to actual current loads $Z_{sr}(n)$, determine their loading status. VP treats undersubscribed (US) contending service routes preferentially over those that are oversubscribed (OS). This preference is implemented through the mechanism of BP at set-up time of new flows associated with the service routes, as explained below. To recapitulate, in the set-up of a new flow on a service route (s, r), each link in the route receives a request for bandwidth allocation d_s, together with the information $\text{status}_{sr}(n)$, where n is the current window.

Bandwidth Protection at link ℓ, which has bandwidth, say, C_ℓ, works as follows. Let $y_\ell(t)(\leq C_\ell)$ denote the total bandwidth usage on the link at the time t of flow set-up. The link implements

Procedure Bandwidth_Protection:

either accept flow

$$\begin{aligned} &\text{if } \{y_\ell(t) + d_s \leq C_\ell \text{ and } \text{status}_{sr}(n) = \text{US}\} \\ &\text{or } \{y_\ell(t) + d_s + R\hat{d} \leq C_\ell \text{ and } \text{status}_{sr}(n) = \text{OS}\} \end{aligned} \tag{15}$$

or else reject flow.

Here $\hat{d} = \max_s d_s$ and R, a small positive number of about 1, is the important *bandwidth protection parameter*. The quantity $(R\hat{d})$ is thus the bandwidth at the link that is set aside to protect, i.e. accommodate, future flows on US service routes.

It is quite remarkable that R as small as 1 or 2 provides a high degree of robustness to

SLA management. The reason for this is that if the link gets overloaded, say by a burst of flows on US service routes, then the bandwidth released by each departing flow contributes to the pool of protected bandwidth. Thus the effectiveness of BP depends on the dynamics of the departure process, the rate of which is high when, as in overload, the number of flows in progress are high.

We note that an attempt to set-up a flow on a selected service route succeeds only if all the links in the route accept the flow after implementing the procedure Bandwidth_Protection.

5.2 Other Control Elements

We briefly review other elements, such as (a) "sticky routes", (b) the notion of "maximally underloaded service route" as determined by an ingress node, (c) recovery time, and (d) selective crankback.

An ingress node σ_1 on receiving a request for a flow on stream (s, σ) has the option of attempting to route it on any route $r \in \mathcal{R}(s, \sigma)$. With sticky routing, the preference is to use the previous service route. If, however, this service route's status has changed to OS or the previous attempt was unsuccessful, then that service route is abandoned and a procedure for selecting a new one is initiated. While stickiness goes counter to load balancing, the consequence of this is not significant in our experience, and outweighed by the gain in route stability, reduced variability and reduced processing. We note that sticky routes were used in [21].

In our route selection algorithm, a measure of last resort is the selection of the maximally underloaded route, which for a stream (s, σ) and time t is defined to be

$$\arg \min_{r \in \mathcal{R}(s,\sigma)} (Y_{sr}(t) - X_{sr}) \tag{16}$$

It is noteworthy that since $Y_{sr}(t)$ and X_{sr} are local to the ingress node σ_1, all the work involved in (16) is performed locally at σ_1.

When an attempt to set-up a flow on a selected service route has failed, the likelihood of the service route being able to accept another request is small immediately afterwards, but improves with time. In our algorithm we do not make another attempt on the service route for a period T_{rec}, which is called the recovery time. A related proposal has been made in [22].

An element that provides linkage from the SLA monitoring processes to the route selection and admission control procedures is selective crankback. Crankback would simply recursively apply the entire process again in the event that an attempt to set-up a flow is unsuccessful, and selective crankback would do so only if the current SLA_state for the stream is "non-compliant" (see Section 2).

5.3 Algorithm

A high level view of the algorithm that is implemented when a request for a flow on stream (s, σ) arrives is illustrated in Figure 2. The ingress node implements the route selection part, before the set-up process involving all the links in the selected route is initiated. In the flowchart the current window is n and also

$$\mathcal{R}_{\text{US}}(s, \sigma; n) = \{r \in \mathcal{R}(s, \sigma) \mid \text{status}_{sr}(n) = \text{US}\} \tag{17}$$

is the set of admissible routes that are currently undersubscribed.

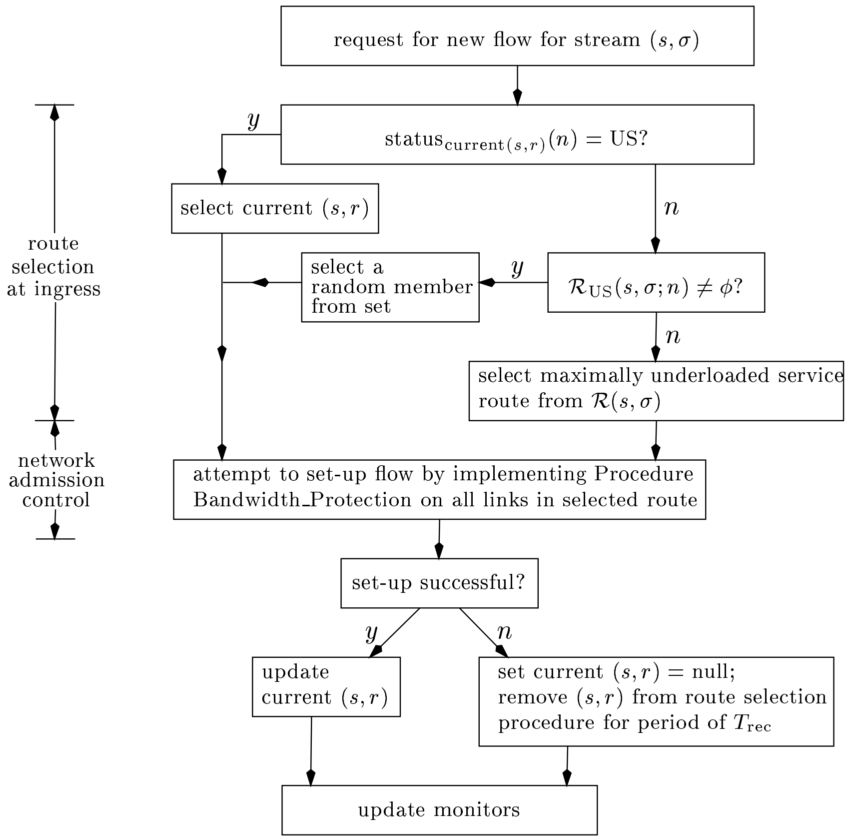

Figure 2: Flowchart of routing and admission control (selective crankback not shown)

6 Numerical Results from Case Study

We first describe the assumptions in the case study before presenting the numerical results.

6.1 Case Study Assumptions

The case study is based on a fictitious network called Core NSF [3], which has eight nodes ($N = 8$), of which 10 pairs are directly connected, as shown in Figure 3. The network has 20 directed links ($L = 20$), one in each direction for each connected node pair. The typical bandwidth of a directed link is OC3 = 155 Mbps, with the exception of the links connecting Argonne ③ and Princeton ④, and also Houston ⑧ and Atlanta ⑦, which have bandwidths of $2 \times$ OC3 = 310 Mbps. One measure of total resources in the network is 24 OC3-hops.

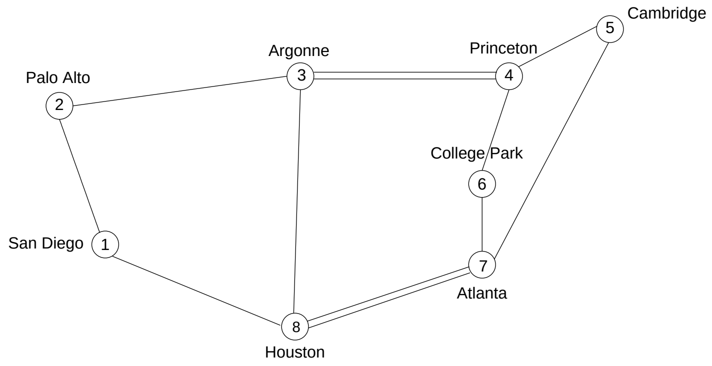

Figure 3: Network for case study

There are six service classes: voice, data 1, data 2, data 3, data 4, video, indexed by $s = 1, 2, \ldots, 6$, respectively. The effective bandwidths of individual flows of these classes are $d_s = 16, 48, 64, 96, 384$ and 640 Kbps. Voice ($s = 1$) and video ($s = 6$) are delay sensitive service classes, and their admissible route sets $\mathcal{R}(s, \sigma)$ consist only of routes with the minimum number of hops. There are a total of 68 routes for each of these two service classes. The four remaining are data service classes, all delay insensitive. Their admissible route sets $\mathcal{R}(s, \sigma)$, $s = 2, 3, 4, 5$, are identical and consist of routes with at most four hops. For each such s there is a total of 160 routes.

The mean duration or holding times, h_s, of flows of the service classes are as follows: $h_s = 1$, 1, 1, 4, 4, 6.67, where the unit of time is 3 minutes. Thus video flows last on average for 20 minutes.

We next describe the aggregated bandwidths offered to streams (s,σ), $U_{s\sigma}$, that are also stipulated in the SLA and used in the design (see Section 2 and 3.1). We define the matrices $\boldsymbol{U}_s = \{U_{s\sigma}\}$, $s = 1, 2, \ldots, 6$, and, furthermore, for compactness we define a single base matrix $\boldsymbol{U}$ from which we obtain $\boldsymbol{U}_s = k_s \boldsymbol{U}$, where k_s is a scalar multiplier. The multipliers are $k_s = 0.39, 0.14, 0.12, 0.14, 0.11, 0.10$. As the reader will observe, the total offered traffic for the real time services ($s = 1$ and 6) are approximately balanced by that for data services. Table 2 gives the matrix $\boldsymbol{U}$.

Table 2: Base matrix $\boldsymbol{U}$, in Mbps

–	14.1	16.5	2.4	21.2	11.8	4.7	7.1
16.5	–	56.6	7.1	73.1	35.4	14.1	21.2
18.9	58.9	–	9.4	87.2	42.4	16.6	25.9
2.4	7.1	7.1	–	9.4	4.7	2.4	2.4
18.9	70.7	84.9	9.4	–	54.2	18.7	30.7
11.8	33.0	37.7	4.7	49.5	–	9.4	14.1
4.7	11.8	14.1	2.4	18.9	9.4	–	4.7
7.1	18.9	23.4	2.4	28.3	14.1	4.7	–

The reader may verify that the sum of all entries in the base matrix is 1209.2 Mbps. For the experiments the conversion from carried flows to revenue is calculated on the basis that 16 Kbps bandwidth carried for a unit of time generates unit revenue. Hence, if all the bandwidth specified in the base matrix is offered to the network and carried, the corresponding rate of revenue generation is 7.56×10^4.

The design for the case study was done by the techniques described in [3] and incorporated in the software package TALISMAN. The design gives the flow acceptance ratios for individual streams that exceed .99.

6.2 Experimental Results

We evaluate the various elements of the SLA management scheme that has been described above. This is done through experiments on D'ARTAGNAN (**D**esign **A**ssisted **R**eal **T**ime **A**l**g**orithms for **N**etwork Service Level **A**greeme**n**ts). In the experiments we consider three scenarios, each with a distinctive traffic pattern that is characterized by the set of actual offered aggregate traffic for all streams (s,σ), i.e., for all service classes and ingress-egress node pairs. The traffic patterns are:

Table 3: NORMAL traffic scenario

R, Bandwidth Protection	Revenue per unit time ($\times 10^4$)	Penalty per unit time $m_s = 1$ ($\times 10^4$)	Net Revenue per unit time ($\times 10^4$)		
			$m_s = 1$	$m_s = 5$	$m_s = 10$
0	7.46024	0.00775	7.452	7.421	7.383
1	7.45086	0.00585	7.445	7.422	7.392
2	7.44299	0.00616	7.437	7.412	7.381
3	7.43479	0.00656	7.428	7.402	7.369

Table 4: BALANCED ABNORMAL traffic scenario

R, Bandwidth Protection	Revenue per unit time ($\times 10^4$)	Penalty per unit time $m_s = 1$ ($\times 10^4$)	Net Revenue per unit time ($\times 10^4$)		
			$m_s = 1$	$m_s = 5$	$m_s = 10$
0	6.97299	0.21680	6.756	5.889	4.805
1	6.87995	0.01519	6.865	6.804	6.728
2	6.87025	0.00248	6.868	6.858	6.845
3	6.86073	0.00486	6.856	6.836	6.812

Table 5: UNBALANCED ABNORMAL traffic scenario

R, Bandwidth Protection	Revenue per unit time ($\times 10^4$)	Penalty per unit time $m_s = 1$ ($\times 10^4$)	Net Revenue per unit time ($\times 10^4$)		
			$m_s = 1$	$m_s = 5$	$m_s = 10$
0	8.43146	0.65547	7.776	5.154	1.877
1	8.28662	0.09143	8.195	7.829	7.372
2	8.22821	0.03907	8.189	8.033	7.838
3	8.19727	0.04961	8.148	7.949	7.701

(i) NORMAL: The ideal case where the offered traffic $U_{(s,\sigma)}$ is identical to the stipulated quantities in the SLA and Design. The results for this case calibrate the expectations in the SLA, and provide a benchmark for the performances of the SLA management algorithm in the other, abnormal cases.

(ii) BALANCED ABNORMAL: Half the node pairs, which are selected arbitrarily, have no offered traffic at all, while the other half have offered traffic for each of the service classes which are twice the SLA/Design values. Note that in this case, the total offered traffic to the network is about the same as in the preceding "normal" case, hence the term "balanced".

(iii) UNBALANCED ABNORMAL: 25% of all node pairs, which are selected arbitrarily, have actual offered traffic for each of the service classes which are twice as much as their respective values in the SLA/Design, while for the remaining 75% the actual offered traffic is as expected. Note that in this case the network as a whole has about 25% more offered traffic than in the preceding cases, and may be considered to be in general overload.

Traffic patterns (ii) and (iii) correspond to important situations in which contentions exist between customers behaving predictably and the remaining, overloading customers. These cases are of particular interest in the evaluation of the algorithms, and the ability to detect the contention and respond appropriately by carrying as much offered traffic as is advisable for the given penalty clauses and parameters in the SLA.

The lifetimes or holding times of the flows are assumed to be either exponentially distributed, or, as in the last subsection, to have the heavy-tailed Pareto distribution with finite mean and infinite variance.

While net revenue, $W\_\mathrm{net}(\cdot)$, and penalty $(\cdot)$ have been defined to be cumulative (see (2) and (3)), the results presented in this section are for unit time, i.e., obtained from the cumulative quantities by dividing by the length of the simulated time. These quantities are obtained for each traffic pattern and various values of the SLA/Design and control parameters. The revenue calculations are on the basis of actual carried bandwidth, as already discussed. The base penalty is calculated on the assumption that the penalty multiplier, m_s, is unity. The net revenue rates are calculated for $m_s = 1, 5, 10$. It is important to recall that in contrast to the results reported here which are highly aggregated, the simulation testbed provides time dependent revenue and penalty data at the ingress node for each stream that originates there.

It is noteworthy that the sample path (time and profile of every flow request) is identically reproduced for all the experiments in a given scenario. For every experiment 10 million flows are simulated, and the statistics reported here are based on results collected after a transient period, which is chosen to be sufficiently large for steady state to be reached. The number of flows that contribute to the statistics is sufficiently large to make the confidence intervals negligibly small.

The parameters of interest in this study are β, the compensation parameter in the Design/SLA interface; α and τ, the exponential smoothing parameter and window length in the measurement process, and, importantly, R, the bandwidth protection parameter.

The measurement parameters have been chosen empirically. A larger α implies greater smoothing, just as a larger window length does. Increasing either one improves the quality of the measurement but at the cost of a slower response to significant traffic fluctuations. In our experiments we have found that a satisfying compromise is to set τ equal to unity, the order of the average holding time, and to have α of 0.8. Also, for the results reported here we have taken the smoothing parameter and window length in the SLA monitoring process to be the same as above.

6.2.1 Effect of the Bandwidth Protection

The effect of the bandwidth protection on the net revenue are indicated in Tables 3, 4 and 5 for normal, balanced abnormal and unbalanced abnormal scenarios, respectively. For the experiments we fixed the parameters γ and β to 0.5. Here we do not apply the selective crankbacks and recovery time mechanisms.

For normal traffic conditions, the effect of the bandwidth protection and the penalty multiplier on the net revenue is small. This is expected because the routing algorithm is optimized specifically for this traffic condition so as to maximize the revenue and also the SLA has been crafted so that the actual carried bandwidth is very close to the offered bandwidth, indicating a small loss ratio. As a consequence the penalty is insignificant in comparison to the total generated revenue. Observe also that the generated total revenue decreases slightly as we increase the bandwidth protection. This behavior indicates that bandwidth protection is being applied even in normal condition because of the bursty nature of the offered traffic. SLA crafting by way of appropriate setting of the parameter γ (see

(9)) can tune this effect.

Turning next to the balanced abnormal traffic pattern, for the first time we observe a noticeable gap between the offered bandwidth and the actual carried bandwidth, even though the total offered bandwidth is close to normal. Now most important is the effect of the bandwidth protection; while the protection does not induce a dramatic loss in terms of total generated revenue, the penalty is reduced by one order of magnitude when one unit of bandwidth protection is applied and by another half when two units of bandwidth protection are applied. In the case of unbalanced abnormal traffic, this behavior is accentuated, and in both scenarios we see that a small protection is surprisingly beneficial and sufficient. Depending on the penalty multiplier used, our results indicate that an optimal value for the bandwidth protection parameter is either 1 or 2.

6.2.2 Effect of Compensation Parameter in Design-SLA Interface

Table 6 illustrates the effect of varying β for the three scenarios when the bandwidth protection parameter $R = 1$, the other parameters being the same as above.

Table 6: Effect of β for each traffic scenario

Traffic Scenario	β	Revenue ($\times 10^4$)	Penalty $m_s = 1$ ($\times 10^4$)
Normal	0.0	7.44299	0.00924
	0.5	7.44299	0.00616
Balanced Abnormal	0.0	6.87025	0.00710
	0.5	6.87025	0.00248
Unbalanced Abnormal	0.0	8.22821	0.07051
	0.5	8.22821	0.03907

6.2.3 Effect of Selective Crankbacks

Table 7 illustrates the effect of selective crankbacks.

The results indicate that the improvement obtained by selective crankback is marginally profitable in net revenue, but does not warrant the additional burden in the call set-up processing. For example for the unbalanced abnormal scenario in Table 7, where $R = 1$, increasing the crankback from 0 to 1 only gives a 0.2% improvement in net revenue with

Table 7: Effect of selective crankback

Crankback	Revenue ($\times 10^4$)	Penalty $m_s = 1$ ($\times 10^4$)	Net Revenue $m_s = 1$	$m_s = 5$ ($\times 10^4$)	$m_s = 10$
0	8.28662	0.09143	8.195	7.829	7.372
1	8.29465	0.08145	8.213	7.887	7.480

penalty multiplier $m_s = 1$. We also observed in some of the other experiments, the results of which are not reported here, that increasing the crankback actually decreases the net revenue, especially when the bandwidth protection is higher. It clearly indicates that, at least for this case study, selective crankback is not worthwhile.

6.2.4 Effect of Holding Time Distribution

In all the experiments presented above the accepted flows last for an exponentially distributed random time with specified mean values. The following experiment focusses on the effect of the holding time distribution on our SLA management scheme. For this purpose we now assume that the duration of the video class (6) flows are independently drawn from a Pareto distribution [23]. That is, flows of class 6 last for t units of time with probability density

$$P(t) = ab^a / t^{(a+1)},$$

if $t \geq b$ and $P(t) = 0$ if $t < b$ ($a > 0$, $b > 0$). The Pareto belongs to the class of heavy tailed distributions, i.e., the decay of its density function for large t is slower than exponential. The mean is $ab/(a-1)$ and it is infinite if $a \leq 1$. If $1 < a \leq 2$, the mean is finite but the second moment is infinite. In this experiment we set $a = 2$, so that the standard deviation is infinite. We select b such that the mean is as before for the exponentially distributed holding times, i.e., $b = h(\text{video})/a$. We also fix the bandwidth protection parameter, $R = 2$, with the other parameters same as in Section 6.2.1. The results in Table 8.

It is impressive that the scheme is not sensitive to the holding time distribution. Note that the video class accounts for approximatively 10% of the total offered traffic.

Table 8: Effect of holding time distribution

Traffic Scenario	Holding Time Distribution	Revenue ($\times 10^4$)	Penalty $m_s = 1$ ($\times 10^4$)
Normal	exponential	7.44299	0.00616
	pareto	7.43653	0.00758
Balanced Abnormal	exponential	6.87025	0.00248
	pareto	6.87396	0.00326
Unbalanced Abnormal	exponential	8.22821	0.03907
	pareto	8.20530	0.04247

7 Conclusion

We have proposed a template for QoS-centered Service Level Agreements and a framework for its real time management in multiservice packet networks. We have also evaluated the SLA management scheme in a case study performed on D'ARTAGNAN, a simulation testbed. The results show that the scheme is robust, fair and efficient over a broad range of traffic conditions.

Current work is directed at the management of Service Level Agreements for multiple Virtual Private Networks supported on a service provider's infrastructure. This work exploits the design techniques for VPNs in [5] and [6], as well as the resource sharing techniques in [11].

References

[1] F. P. Kelly, "Routing in circuit-switched networks: optimization, shadow prices and decentralization," *Adv. Appl. Prob.*, vol. 20, pp. 112–144, 1988.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1992.

[3] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM network design and optimization: A multirate loss network framework," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 531–543, 1996.

[4] A. Shaikh, J. Rexford, and K. Shin, "Efficient precomputation of Quality-of-Service routes," *Proc. NOSSDAV 98,* Cambridge, England, July 1998.

[5] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "Virtual private networks: Joint resource allocation and routing design," *Proc. IEEE INFOCOM 99*, pp. 480–490, 1999.

[6] D. Mitra and K. G. Ramakrishnan, "A case study of multiservice, multipriority traffic engineering design for data networks," *Proc. GLOBECOM 1999*, vol. 1B, pp. 1077–1083, Brazil.

[7] M. Montgomery and G. de Veciana, "Hierarchical source routing through clouds," *Proc. IEEE INFOCOM '98*, pp. 685–692, 1998.

[8] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 365–386, 1995.

[9] D. Mitra and I. Ziedins, "Virtual partitioning by dynamic priorities: Fair and efficient resource sharing by several services," in *Broadband Communications, Proceedings of the International Zurich Seminar on Digital Communications*, B. Plattner, Ed. 1996, pp. 173–185, Springer.

[10] S. C. Borst and D. Mitra, "Virtual partitioning for robust resource sharing: Computational techniques for heterogeneous traffic," *IEEE J. Sel. Areas Communications*, vol. 16, pp. 668–678, June 1998.

[11] D. Mitra and I. Ziedins, "Hierarchical virtual partitioning: Algorithms for virtual private networking," *Proc. IEEE GLOBECOM'97*, pp. 1784–1791, 1997.

[12] Q. Ma and P. Steenkiste, "Supporting dynamic inter-class resource sharing: A multi-class QoS routing algorithm," *Proc. IEEE INFOCOM 99*, pp. 649–660, 1999.

[13] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, Nov./Dec. 1997.

[14] *IEEE J. Selected Areas in Communications*, vol. 13, no. 6–7, Aug.–Sept., 1995.

[15] M. Schwartz, *Broadband Integrated Networks*, Prentice Hall, New York, 1996.

[16] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource ReSerVation Protocol," *IEEE Network*, vol. 7, pp. 8–18, September 1993.

[17] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A framework for QoS-based routing in the Internet," Work in progress, draft-ietf-qosr-framework-0.5-txt, May 1998.

[18] W. C. Lee, M. G. Hluchyj, and P. A. Humblet, "Routing subject to quality of service constraints in integrated communication networks," *IEEE Network*, vol. 9, no. 4, pp. 14–16, July 1995.

[19] D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," Work in progress, draft-ietf-mpls-traffic-eng-00.txt, April 1999.

[20] V. E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*, chapter 6, Academic Press, 1965.

[21] R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic alternative routing – modeling and behavior," *Proc. 12th Int. Teletraffic Congress,* Torino, pp. 3.4A.3.1–3.4A.3.7, 1988.

[22] N. G. Bean, *Statistical Multiplexing in Broadband Communication Networks*, Ph.D. dissertation, Univ. Cambridge, U.K., 1993.

[23] J. Galambos, *Advanced Probability Theory*, Marcel Dekker, 1988.