A SLA Framework for QoS Provisioning and Dynamic Capacity Allocation

Rahul Garg
IBM India Research Lab
New Delhi, INDIA
Email: grahul@in.ibm.com

Ramandeep Singh Randhawa
Graduate School of Business
Stanford University, CA, USA
Email: rsr@Stanford.EDU

Huzur Saran
Dept. of Computer Science and Engineering
Indian Institute of Technology
New Delhi, INDIA
Email: saran@cse.iitd.ernet.in

Manpreet Singh
Dept. of Computer Science
Cornell University
Ithaca, New York, USA
Email: manpreet@cs.cornell.edu

***Abstract*—**
Any QoS scheme must be designed from the perspective of pricing policies and service level agreements (SLAs). Although there has been enormous amount of research in designing mechanisms for delivering QoS, its applications has been limited due to the missing link between QoS, SLA and pricing. Therefore the pricing policies in practice are very simplistic (fixed price per unit capacity with fixed capacity allocation or pricing based on peak or 95-percentile load etc.). The corresponding SLAs also provide very limited QoS options. This leads to provisioning based on peak load, under-utilization of resources and high costs.

In this paper we present a SLA based framework for QoS provisioning and dynamic capacity allocation. The proposed SLA allows users to buy a long term capacity at a pre-specified price. However, the user may dynamically change the capacity allocation based on the instantaneous demand. We propose a three tier pricing model with penalties (TTPP) SLA that gives incentives to the users to relinquish unused capacities and acquire more capacity as needed. This work may be viewed as a pragmatic first step towards a more dynamic pricing scenario.

We solve the admission control problem arising in this scheme using the concept of trunk reservation. We also show how the SLA can be used in virtual leased-line service for VPNs, and web hosting service by application service providers (ASPs). Using web traces we demonstrate the proposed SLA can lead to more efficient usage of network capacity by a factor of 1.5 to 2. We show how this translates to payoffs to the user and the service provider.

I. Introduction

There has been enormous research in defining Quality of Service (QoS) of a resource serving multiple users [9], [2] and designing mechanisms to provide the QoS [23], [2], [31], [10], [30], [28], [1]. Similarly, there is a significant body of research literature designing pricing policies for the usage of these resources [20], [22], [21], [6], [24], [25].

Much of this research remains unused in practice because of lack of QoS demands from users, lack of willingness of service providers to adapt complex pricing mechanisms, and the missing link between the QoS, service level agreements (SLAs) and pricing. For an average user the complex QoS metric do not lead to a significant advantage. In general, the users do not know how to efficiently map their performance requirements to a complex QoS metric. Moreover, many of the sophisticated QoS and pricing mechanisms are complex to implement and therefore infeasible. Most of the proposed approaches are at a significant departure from the currently installed infrastructure and currently practiced pricing policies. Therefore their adaptation is difficult, even if they are better.

As customers have begun to demand higher level of Quality of Service (as opposed to the *best effort* service) from the service providers (especially from long distance carriers, ISPs, and ASPs), service level agreements (SLAs) between customers and service providers have become the norm. These SLAs specify the quality of service and pricing information.

For instance in a SLA between a customer and a frame-relay based bandwidth provider, typical QoS metrics included are, committed bandwidth, transit delay, packet (or cell/frame) loss rate and availability. Most of the frame-relay providers are converging to the same QoS parameters of 99.99 percent availability, 0.001 percent loss rate and 50-100 ms of one way transit delays. Therefore the pricing of these services are mainly based on the fixed committed bandwidth (committed information rate) requested by customers [8], [27].

In the context of IP networks, Internet service providers (ISPs) are not able to guarantee any availability and loss rate parameters. These ISPs often price the service according to a 95-5 model. In this model, the customer buys a committed bandwidth from the ISP at a fixed rate. However, the customer is allowed to send traffic at rates higher than the committed bandwidth. The ISP measures the average bandwidth usage of each customer in every five minute period by measuring the total traffic sent or received by the customer in every five minute interval. These average bandwidth measurements of a customer are accumulated over a period (of typically a month) and then sorted. The highest five percent of the samples are discarded to compute the 95th percentile of these samples. The difference between the so computed 95 percentile and the committed

This work was done when Manpreet Singh, R.S. Randhawa and Prof. Huzur Saran were visiting IBM India Research Lab.

bandwidth is charged at a different (higher) rate. Similar 95-5 models are also used by ASPs to price the bandwidth used by their customers.

Both, the leased line style fixed capacity model and the ISP style 95-5 pricing model are based on the peak consumption of the user. This leads to over-provisioning of capacity and under-utilization of resources, leading to high costs for the customers and low revenues to the service providers.

More recently researchers have began to investigate simpler approaches for QoS provisioning. For instance Duffield et al. [7] describe a capacity resizing approach that allows the users to dynamically change their guaranteed bandwidth allocation depending on their requirements. Using the ATT call data they show that dynamic resizing of VPN pipe capacities can result in up-to factor 2 savings in the provider network capacity. The work in [11] proposes a scheme to carry out max-min fair sharing of provider capacities in case of overload. Such approaches are easy to implement in most of the communication technologies using appropriate signaling protocols like RSVP-TE and CR-LDP [4], [18] for IP networks, UNI and PNNI [2], [3] for ATM networks, and frame-relay signaling for frame-relay based networks.

In this paper, we propose a SLA based on a three tier pricing policy with penalties (TTPP) for the capacity resizing model. In the proposed approach, a user books a long term provisioned capacity at a negotiated price. However, the user can dynamically change its capacity allocation depending upon its resource requirements. The net payment of the user depends on the actual capacity allocation. When a user gives up some of its provisioned capacity, it is entitled to a *discount*. Similarly, when a user requests for additional capacity and is allocated a capacity larger than its provisioned capacity, the user is charged for the additional capacity at a *premium* price. Our framework also allows the user to quickly reclaim capacity which has been given up earlier to obtain a discount. In case the service provider is not able to give back the capacity, it is required to pay an appropriate *penalty* for that period. The crux of this approach is that the dynamic resizing of the allocated capacities is done by software agents acting on behalf of users. When the service provider gets a request to increase the capacity allocation of a user, it needs to decide, in an automated manner, whether to accept or reject the request.

We study this admission control problem from the perspective of revenue maximization of the service provider. We give a trunk-reservation based admission control process for admitting the resize requests. We demonstrate using actual (web) data traces that our scheme works well thereby validating our assumptions. The scheme has low overheads and results in significant payoffs both for the service provider and the users. We demonstrate how the proposed SLA may be used by ASPs in pricing and allocating their resources, and by VPN service providers in pricing and allocating the bandwidth to their customers.

The proposed SLA is evolutionary in nature and can co-exist with the current fixed capacity model: users not interested in the complexity of resizing can choose not to resize and continue to operate at the fixed capacity model, while some others may choose simple time of day based resizing, and more advanced users can bring to bear sophisticated resizing techniques to get the full benefit of dynamic capacity allocation.

The rest of the paper has been organized as follows. We describe the proposed SLA in Section II. In Section III we describe the admission control problem faced by the service providers and suggest a trunk-reservation based admission control algorithm. In Section IV we demonstrate how ASPs can use the proposed SLA in pricing and allocating their resources to their customers. Section V describes the application of the proposed SLA in VPNs. We present some preliminary simulation results in Section VI to demonstrate the potential gains to the users and the service providers while using the proposed capacity allocation and pricing policy. We conclude in Section VII.

II. The Three Tier Pricing Policy with Penalties (TTPP) SLA

Consider a resource shared among multiple users. Every user signs a service level agreement (SLA) with the provider of the resource (also called the service provider). The SLA includes a QoS specification and a pricing policy.

The QoS specified in a SLA may be divided into two parts: static QoS specifications, and dynamic QoS specifications. The static QoS specifications include parameters that are fixed and are expected to remain unmodified during the lifetime of the SLA. These parameters include reliability, availability, mean time before failure, grade of service (premium, gold, bronze etc.), (packet) loss rate etc., of the resource for a user. The dynamic part of QoS specifications include the parameters that the users and the service providers would like to modify dynamically during the lifetime of the SLA.

We consider the simplest case where the dynamic part of the QoS specifications is represented by a single capacity parameter $u(t)$, representing the amount of resource allocated to the user at a given time t. This may represent the *committed information rate* in a frame-relay SLA, or a guaranteed bandwidth provided in a *virtual leased line* service, or the number of server machines (or bandwidth) allocated to a web-site hosted at an application service provider (ASP). Most of the SLAs currently in practice keep the allocated capacity fixed leading to over-provisioning of the resource by user and lower resource utilization of the service provider. Our proposed SLA allows the users to dynamically change their capacity allocations depending upon their instantaneous requirements.

The proposed SLA between a user and a service provider consists of:

- A specification of the static QoS parameters,
- A long-term expected capacity requirement C (called the provisioned capacity),

- A charging rate r per unit capacity per unit time for the provisioned capacity,
- A discount rate d given to the user when it relinquishes a part of its provisioned capacity,
- A premium rate p, at which the user is charged for capacity allocation beyond provisioned capacity (C),
- A penalty q given by the service provider when it is unable to immediately reallocate the relinquished capacity of the user.

Consider a user signing a SLA for a period T_1 to T_2, with the provisioned capacity C. Let $u(t)$ be the actual capacity allocated to the user at time t. In the absence of penalties, the net amount the user needs to pay at the end of the period (T_1, T_2) would be:

$$rC(T_2 - T_1) + p \int_{T_1}^{T_2} \max(0, u(t) - C)dt$$
$$-d \int_{T_1}^{T_2} \max(0, C - u(t))dt.$$

The capacity allocation by the service provider is non-preemptive. This means that once the provider has allocated a capacity to a user at a premium, it cannot take it back on its own. The provider gets the capacity back only when the user releases it. Preempting capacity allocation may result in service disruption for the user which might be unacceptable (for instance, a user might be using the additional capacity for a real-time video conference).

The pricing parameters of the SLA are expected to be negotiated between the users and the service providers based on business and other considerations. However, one would expect the discount rate d to be lower than the charging rate r. Similarly, one would expect the premium rate p to be larger than r. Moreover, the ratios d/r and p/r may be chosen such that users have sufficient incentives to relinquish their excess capacities when they don't need it, and request additional capacities when their requirements exceed their provisioned capacities.

In general, the discount rates, and premiums could be dynamically adjusted based on dynamic supply-demand scenarios or some auctioning process. For simplicity, and ease of implementation and adaptation, we keep the discount rates and the premium rates fixed.

The service provider provisions its own resource capacity primarily based on the provisioned capacities specified in the SLAs of the users. In addition the service provider may also use other information such as discount rate d, premium rate p and the experiences from the past history to provision its capacity.

The penalty q is present in the SLA for the following reason. Since, typically the premiums will be larger than discounts, at times of resource scarcity, it may be in the interest of the service provider to delay returning resources that has been released by a user. For instance, suppose there are two users A and B with equal provisioned capacities of C. If user A is not using its full capacity, and user B requires extra capacity, then under this arrangement, A may give up some of its capacity to the service provider who in turn sells it to B. As a result the service provider gets an additional revenue of $p - d$ per unit time, per unit of capacity reallocated. However, at a later stage if user A needs its capacity back, under the proposed SLA, there is no incentive to the service provider to return back the capacity. Moreover the service provider has an incentive to sell this capacity to another user at a premium. So, it is important to have a mechanism by which any capacity released may be reclaimed with a short notice and a penalty clause is included to make sure that the service provider makes its best effort to return the borrowed capacity at the earliest.

There are many choices for penalties: *fixed penalty, delay-dependent penalty* and *proportional penalty*.

Fixed penalty: When a user asks for some of its relinquished capacity back and the capacity is not given back to him immediately, then a penalty of q is credited to the user's account. The higher the penalty q, the sooner the service provider would try to return the capacity to the user.

Delay-dependent penalty: The penalty is proportional to the delay incurred by the service provider in returning the capacity. If the service provider returns the capacity immediately, no penalty is due. However, if the service provider waits for another user to release some capacity, then the penalty due is proportional to the difference in time when the user actually gets the capacity back and the time when the user makes the request to get back some of its capacity. If q is the agreed penalty rate in the SLA and t_{req} and t_{alloc} are the time instants when the capacity was requested and the capacity was actually allocated respectively, then the service provider's penalty due to the user is $q(t_{alloc} - t_{req})$.

Proportional penalty: This is also a form of delay-dependent penalty, where the penalty to be credited to a user is also proportional to the difference between the user's provisioned capacity C and its current allocation $u(t)$. If q is the agreed penalty per unit capacity per unit time, and t_{req} and t_{alloc} are the respective times when the capacity was requested and allocated, then the amount credited to the user by the provider is $q(C - u(t_{req}))(t_{alloc} - t_{req})$.

In all of these cases, if a user requests extra capacity at a premium and the service provider is unable to allocate it, no penalty is due, as the service provider is not obliged to provide extra capacity. Any combination of these three types of penalties may be used. For simplicity we have studied only fixed penalties in the rest of this paper.

III. Admission Control: The Problem and its Solution

To alter the capacity allocations, software agents acting on the behalf of users send increase/decrease (resize) messages to the service provider. In the TTPP SLA, if a user sends a request to decrease its capacity allocation, the request can always be accepted by the service provider. However, when the user sends a request to increase its capacity allocation, a decision has to be made by the service provider whether to accept or reject the request. Since the capacity allocations are non-preemptive, once a request is accepted, the new allocated capacity cannot be taken back from the user unless the user voluntarily releases it. If the available capacity is small, then accepting a request to increase allocation may potentially force the service provider to reject a future request of higher premium by another user. The service provider may also have to pay penalty to another user whose current usage is less than the provisioned capacity. In the fixed penalty model, the service provider cannot queue the increase request and has to take this decision at the instant the request arrives. This admission control decision is complex. For the service provider, it is desirable to design the admission control policy that maximizes its total revenue. In case the request of a user cannot be accepted, we say that it is blocked. The probability of occurrence of such an event is termed as blocking probability(P^b).

Similar admission control problems have been studied in the context of telecommunication networks [19], [13], [12], [5], [26] where trunk reservation based schemes have been designed to work well. We first describe the concept of trunk reservation and then show how it can be used for admission control for the proposed TTPP SLA.

Suppose there are N users sharing a resource of total capacity of C units. Let every user send request to increase or decrease its capacity allocation by a fixed amount b ($b \ll C$). Suppose each request to increase or decrease allocation arrives randomly. A trunk reservation scheme defines a trunk reservation parameter tr_i *against* every user i. The algorithm tries to ensure that at least tr_i amount of resources are kept available for handling requests of other users. So, according to the trunk reservation policy, a request of a user i is accepted if and only if the capacity of free resources after accepting the request is at least tr_i. Let the amount of resources used by user j at time t be denoted by $u_j(t)$. A new request of user i to increase the capacity allocation by b units, is accepted if and only if:

$$\sum_{j=1}^{N} u_j(t) + b + tr_i \leq C. \tag{1}$$

It has been shown in the context of telecommunications networks [26], [5] that if $b \ll C$ then, even a small amount of trunk reservation gives almost absolute priority to one user over the other. It has been found that usually, a small amount of trunk reservation (as compared to the capacity C) is sufficient for optimal performance.

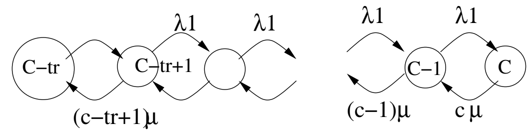

Fig. 1. Markov Chain representing the total state of the system

In the TTPP SLA, since there are four cost parameters including penalty, it is very hard to find the optimal admission control policy. However, a well-designed trunk reservation based policy is still expected to give good results. Even within the trunk reservation policies, it is very difficult to find a closed-form expression for optimal trunk reservation parameters. Since optimal trunk reservation parameters are usually small, a slight over-estimation of trunk reservation is still expected to give good results. We therefore propose to use a trunk reservation based admission control heuristic to decide which capacity-increase requests to accept. We describe this heuristic in the next section.

A. A heuristic for trunk reservation parameter

We define the trunk reservation parameter $\hat{tr}_i$ *for* user i as the amount of resources that we need to reserve in order to be able to handle its future requests. We will first describe how to compute $\hat{tr}_i$ for a user and then show how it can be used to compute tr_i (the trunk reservation against a user) using a priority based scheme.

For computing the trunk reservation parameter, we consider a simplified scenario, as in telecommunication networks. We assume that each resize request is of unit capacity, and requests from different users are independent and form a Poisson process. Requests of user i have a mean arrival rate of λ_i and exponentially distributed service time with a mean of $1/\mu$.

1) Base Case: Consider for simplicity the case of two users sharing a single link of total capacity C units. Let the provisioned capacity of user i be C_i. Consider the case when the charging rate of user i (i = 1 or 2) is r_i. Without loss of generality, assume $r_1 > r_2$. Since service provider's aim is to maximize the overall revenue, we must assign a high priority to user 1. The trunk reservation parameter $\hat{tr}$ for user 1 should be chosen in such a way that the total revenue is maximized.

It can be theoretically proven that, as λ_2 increases the optimal trunk reservation parameter increases. We model this system as a Markov chain with states in $[0 \ldots C]$, where the state k represents that k units of the capacity have been allocated to the users. We compute an approximate upper bound on the optimal trunk reservation parameter, in the limiting case when $\lambda_2 \rightarrow \infty$. Note that this upper bound also holds for all other values of λ_2. In this case, the state of the system will always lie between $C - \hat{tr}$ and C. The moment the state of the system goes below $C - \hat{tr}$, user 2 would immediately take up the excess capacity freed. The Markov chain in Figure 1 shows the transition rates between different states. From Figure 1, when the trunk reservation parameter is $\hat{tr}$, the expression for the blocking probability ($P^b(\hat{tr})$) for requests of user 1 can be computed as follows:

$$P^b(\hat{tr}) = ((\lambda_1/\mu)^{\hat{tr}}/C!)/(\sum_{i=0}^{\hat{tr}} (\lambda_1/\mu)^i/(C - \hat{tr} + i)!). \quad (2)$$

This expression can be simplified as:

$$P^b(\hat{tr}) = (\rho^{\hat{tr}})/(\sum_{i=0}^{\hat{tr}} \rho^i \prod_{j=0}^{\hat{tr}-i-1} (1 - j/C)), \quad (3)$$

where ρ is given by $\lambda_1/\mu C$.

Normally trunk reservation is much smaller compared to C, therefore in the above expression, j/C can be neglected as compared to 1. Thus,

$$P^b(\hat{tr}) \approx \frac{(1-\rho)\rho^{\hat{tr}}}{1 - (\rho^{\hat{tr}+1})}. \quad (4)$$

In the above equation, if ρ is less than 1, then $\rho^{\hat{tr}+1}$ can be neglected compared to 1. Thus,

$$P^b \approx (1-\rho)\rho^{\hat{tr}}. \quad (5)$$

The optimal trunk reservation parameter should be such that the rate of loss of revenue of the second user's requests is just equal to the rate of gain of revenue from the first user's requests. Thus,

$$r_1\mu[P^b(\hat{tr}) - P^b(\hat{tr}+1)] = r_2\mu.$$

Substituting P^b from Eq.(5), we get

$$\hat{tr} \approx \frac{log(r_1/r_2)}{log(1/\rho)} + \frac{2log(1-\rho)}{log(1/\rho)}. \quad (6)$$

The above equation shows that the dependence of trunk reservation on revenues is logarithmic in nature. Assuming that users provide a good estimate of their expected capacity requirements, ρ is expected to be close to C_1/C. Therefore $\hat{tr}$ can be approximated as:

$$\hat{tr} \approx k\frac{log(r_1/r_2)}{log(C/C_1)}, \quad (7)$$

where $k \in [2, 3]$ is a small number chosen to ensure that the trunk reservation used is larger than the optimal.

2) General Case: Now, consider a more general scenario of the TTPP SLA, with revenues, discounts, premiums and penalties. We proceed by first defining the priorities for all the users and then compute the trunk reservation parameter using which these priorities can be implemented efficiently.

If a user's resource usage is less than the capacity specified in the SLA ($u_i(t) \leq C_i$), then accepting an increase request could earn the service provider additional revenue at its discount rate (d_i). Similarly, if the usage is more than the capacity specified in the SLA ($u_i(t) \geq C_i$), accepting an increase request will generate revenue at the user's premium rate (p_i). As the penalties are paid only once for each rejected request, we can say that we would pay penalty to a user at the rate of its arrival. To maximize the net revenue, the users generating more net revenues per unit capacity per unit time should be given higher preference in resource allocation. For this, we define the priority for the i^{th} user as follows:

$$\begin{aligned} Priority(i) &= d_i + q_i\lambda_i \;\; if\, u_i(t) < C_i & (8)\\ &= p_i \qquad if\, u_i(t) \geq C_i & (9)\end{aligned}$$

We sort the users in decreasing order of their priorities and assign a suitable trunk reservation parameter against each user. Note that the trunk reservation against a user with a higher priority should be lower as compared to the trunk reservation against another user with a lower priority. Also note that there should be no trunk against the highest priority user (i.e. all its requests should be accepted as long as there is capacity with the service provider). We extend the form of trunk reservation from Eq.(7) and get the following heuristic expression for $\hat{tr}_i$, the trunk reservation *for* user i :

$$\hat{tr}_i = min(max(0, C_i - u_i(t)), \lceil k\frac{log(\frac{Priority(i)}{Priority(i+1)})}{log(C/C_i)}\rceil) \quad (10)$$

The trunk reservation parameter against a user i (tr_i) is defined as:

$$tr_i = \sum_{j<i} \hat{tr}_j \quad (11)$$

Next, we consider some limiting cases that may arise in TTPP SLA, and show how our trunk reservation based admission control algorithm converges to the optimal scheme for these cases. First, consider the situation where there are no penalties for any user and the revenues, discounts and the mean service times for all are equal, then the optimal policy is that which accepts a request on first-come-first-serve basis (FCFS). We observe that substituting the above mentioned conditions in Eq.(10) and (11) result in zero trunk reservation against all users, i.e. our scheme converges to FCFS. Now, consider the case when there is a very large penalty for every user, and the discount rates, premium rates and revenue rates are all equal. In this case the optimal policy is to never allow any user to resize its capacity beyond its provisioned capacity. In this case from Eq.(10) and (11), we get the trunk reservation against a user i for which $u_i(t) \geq C_i$ is infinity. Therefore, this policy is same as the static provisioning policy where a user is never allocated a capacity that exceeds its provisioned capacity. If user i has a very high arrival rate as compared to the other users, i.e. $\lambda_j/\lambda_i \rightarrow 0\ \forall j \neq i$, the optimal policy should reserve the unused portion of the provisioned capacity for this user. This can be verified from Eq. (10) and (11). Also, if any of the factors, viz. revenue, discount, premium or penalty, increase for a user, the trunk reservation against that user should decrease. This can also be verified from Eq.(10) and (11).

We now describe applications of the proposed SLA in two potential application areas. One application is resource allocation and pricing for application service provider (ASP) services and the other application is resource allocation and pricing virtual private network (VPN) services.

IV. TTPP for Application Service Providers

With the popularity of the Internet, a number of application service providers (ASPs) have emerged. The basic service provided by an ASP is the web hosting service. Currently, the ASPs allocate fixed number of resources (bandwidth and servers) to their customers, thereby limiting their QoS options. However, it is well known that there is high burstiness and unpredictability in the load (number of hits per second experienced by a web-site) of a customer. Therefore, customers have to over-provision their resources corresponding to their expected peak loads, leading to under-utilization of resources and high costs. Even with a significant over-provisioning, a website may not be able to serve all its requests during the periods of unexpected high loads. The charging models of the ASPs are also quite primitive and are primarily based on a flat-fee, amount of data hosted, amount of traffic served, and the peak load.

In the future, ASPs will make use of the potential statistical multiplexing gains among multiple customers, arising due to uncorrelated access patterns of different web-sites, and unexpected and large burstiness in their access patterns, to lower their costs, increase their resource utilization and revenues. These ASPs will provide customers with the option of dynamically allocating and releasing resources depending on their current requirements. Customers, in such a scenario, will not have to provision and pay for the resources for their expected peak load. Instead, they can provision for their nominal load and request and release the resources dynamically as their load changes. The ASP can also realize greater revenues and surplus by providing flexible QoS options to customers, serving more customers with better QoS, while using lesser resources.

The ASPs can use the proposed TTPP SLA for QoS provisioning, dynamic resource allocation and pricing, in such a scenario. With this model, a customer will sign up for a certain number of servers, and/or bandwidth (say C) at a price of r per unit capacity per unit time. When the load on the customer's web site is low (eg. in the night), some of these resources can be released to get a discount. Similarly the customer can request for allocation of additional servers and/or bandwidth at a premium price when the load is high. The revenues, premiums, discounts and penalties will accrue as per the TTPP SLA described earlier. A software agent at the ASP, running on the behalf of each customer, continuously monitors the load at, and the response times of, the web-site and requests for dynamic reallocation of the resources as desired. The requests are admission controlled by the ASP as described earlier.

This model is especially useful in allocating and pricing for the bandwidth resource, if the cost of bandwidth is a significant component of the total ASP costs (e.g. most countries in Asia and Europe).

V. TTPP for Virtual Private Networks

Corporate networks are shifting from leased lines to Virtual Private Networks (VPNs), using frame relay PVCs, ATM virtual paths or connections, IP-based tunnels and virtual leased lines [17]. We refer to a communication path connecting two sites, using any of these technologies as a virtual leased line. Recently, with the advent of multi-protocol label switching (MPLS) [29] and with the availability of constraint-based routing using label distribution protocol (CR-LDP) [18] and RSVP tunneling extensions (RSVP-TE) [4] to provide tunnels with QoS through an IP-based network, it is also envisaged that many of these services will move to the IP platform [14]. Even traditional circuit switched voice is projected to be carried over IP using VOIP technologies [15], although the exact details of this are still being studied. One strong candidate is to use tunnels with QoS thereby segregating it from other best-effort data traffic.

With the older technologies, it was not practical to dynamically resize the leased lines capacities. Therefore, the customers were forced to estimate their capacity requirements and buy capacity corresponding to their expected peak usage. However, with ATM and frame-relay signaling technologies, and the signaling technology (CR-LDP, RSVP-TE) in IP-based networks, it is straightforward to modify the bandwidth allocation of a virtual leased line without the need for any human intervention. In this context, researchers have already shown that resizing capacities could have significant potential benefits, in terms of overall network capacity [7], [15], [11]. All the above capacity resizing based approaches fit very well into our proposed SLA framework.

With the proposed model, a customer buys a long term bandwidth C, on a virtual leased line. The customer also runs a software agent to monitor the traffic load and initiate capacity resize requests as the load changes. These resize requests are sent using appropriate signaling protocols (UNI [2], P-NNI [3], RSVP-TE [4] etc.) depending upon the underlying technology. The requests are admission controlled at the intermediate routers of the virtual leased line (VLL) service provider network as described in Section III. The revenues, premiums, discounts and penalties accrue as per the SLAs described earlier. This ensures that the customers are able to release bandwidth when they don't need it and get it back when they need it, thereby resulting in higher utilization of the underlying network infrastructure. This should lead to lower costs for users and higher surplus for the service provider.

VI. Simulation Results

We use simulations to evaluate the proposed TTPP SLA in the context of ASPs. We built our own discrete event simulator to carry out the simulations. Although, in the Section III, we assumed that resize requests arrive as a Poisson process with an exponentially distributed service time, for our simulations we use a more realistic traffic model based on traces of web traffic.

A. Simulation Model

Consider an ASP hosting web-sites of multiple customers. Assume that the ASP buys a fixed access bandwidth from an Internet Service Provider (ISP), and allocates it to different web-sites. Also assume that ASP charges its customers based on their bandwidth usage and that the bandwidth cost forms a significant portion of the total cost to a customer.

We compare the TTPP SLA with the 95-percentile provisioning model, where the provisioned capacity ($C_i^{0.95}$) of user i is equal to the 95th percentile of its bandwidth usage. In this model the customer pays a fixed price per unit capacity per unit time for its provisioned capacity. We call this policy as the peak provisioning policy.

We compare the total payments made by a user under TTPP SLA with that made in the peak provisioning policy. To obtain a suitable metric for comparison, we set the provisioned capacity in the TTPP SLA, at a level which gives the same blocking probability ($\approx 5\%$) as in the case of peak provisioning. We set the total capacity C of the ASP as the sum of the provisioned capacity for each user, and run the simulations for TTPP SLA with the admission control algorithm of Section III. We compute the payoff for each user and for the application service provider.

We use traces of actual web traffic in our simulations. These traces were taken from the Internet Traffic Archive [16]. We use five traces of varying duration ranging from a week to a month. The traces contain a record for each hit on the web site. Each record contains a time-stamp (time of request arrival) and the size of the data to be transferred. The time stamps in the traces had a resolution of one second. This leads to an artificial bunching of requests at the beginning of each second. To remove this undesirable bunching, we mapped hits at time t to a random time in the interval $[t, t+1)$. There was no record of the completion time of the requests in the traces. For simplicity we assumed that for each hit, the data is transferred at a fixed rate of 20 Kbps. From these traces we extracted five sub-traces of equal length to model traffic of five customers hosted at the ASP. Figure 2 plots the number of hits per second for one of the customer (customer 2) for a period of 24 hours.

An important issue in the implementation of TTPP SLA is the frequency of resizing. The goal of the simulations was to assess the maximum achievable gain from our proposed SLA. Therefore, we assumed that the user increases its allocation by 20 Kbps at the arrival of the web request and decreases this allocation when the data transfer completes.

All these traces belong to the same time zone. Therefore, the loads in all the traces are correlated because of well-known crests and troughs during the days and nights. This suggests that the ASPs should host web-sites with users from different time-zones. To assess the gains in such situations, we shifted the trace of a web-site by 12 hours while keeping the others unchanged and compared the results with the case when the crests of all the web-sites were synchronized.

The values of the TTPP SLA parameters are shown in Table I. We calculate the priorities using Eqs.(8), (9). We calculate trunk reservation parameter as given by Eqs.(10), (11).

B. Performance Metrics

We measure the benefits of the TTPP SLA over peak provisioning with fixed price payment. The payoff (η_i^C) for i^{th} user is defined as the ratio of the total payment made in the fixed price SLA to that in the TTPP scheme. Similarly, the payoff (η^N) for the ASP is defined as the ratio of the net revenue earned per unit capacity installed under TTPP SLA to that under the peak provisioning scheme.

The division of the payoffs between the service provider and the users is dependent on the pricing parameters of the two SLAs. We assume that the charging rate (r) in the TTPP SLA is same as the price of bandwidth in the peak-provisioning SLA. The other pricing parameters of the TTPP SLA used in simulations, are relative to the charging rate (r) and are shown in Table I.

Parameter	Value
r	1
p	2
d	0.5
q	5

TABLE I
COST PARAMETERS USED IN SIMULATION

C. TTPP vs. Peak Provisioning

All the traces were initially synchronized such that their loads increased at the same time. Figure 3 shows the load of one of the users for a two hour interval. It can be seen from the figure that the TTPP scheme requires a smaller provisioned capacity than the peak provisioning scheme.

A comparison of the provisioned capacity and the blocking probability in the TTPP SLA with the peak provisioning is presented in Table II. Note that the provisioned capacity for each user in TTPP SLA is lower than that in the peak provisioning scheme while the blocking probabilities in the two schemes are comparable. The payoff for the ASP is 1.06. The payoff for each user and the ASP is given in the first column and their respective blocking probabilities are given in the third column of Table III. It can be seen that the payoffs of the users vary from 1.3 to 5.02.

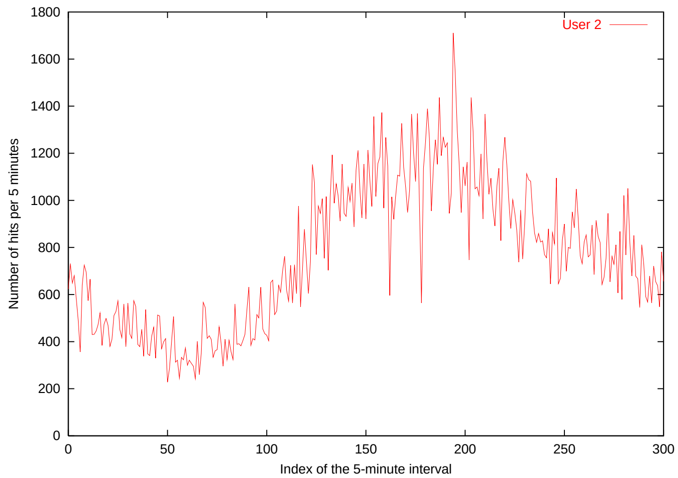

Fig. 2. Variation of hits per second with time

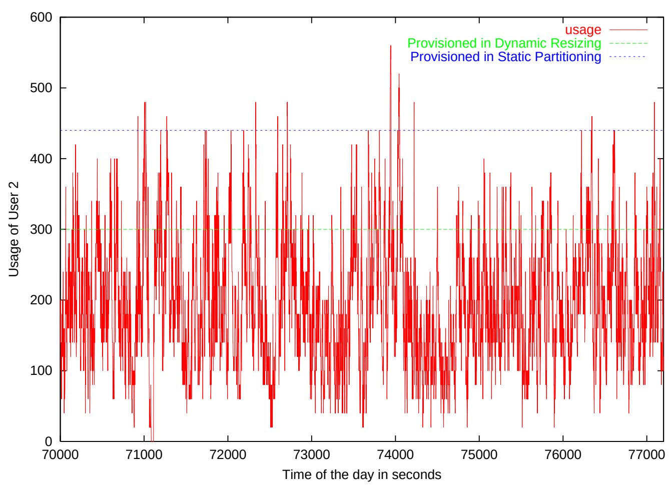

Fig. 3. Variation of the bandwidth used with time

D. Synchronized Loads vs Shifted Loads

Next, we compare the payoffs and blocking probabilities under synchronized loads and shifted loads. The results are shown in Table III. Note that the blocking probability of all the users decreases in the case of shifted loads. Surprisingly, the users get lower payoffs in the case of shifted loads. This is so because, as less number of requests for each user are now turned down, the actual payment made by each user is more. Moreover, in this case, the users required smaller provisioned capacities and often resized it beyond their provisioned capacities. This reduces the payoff for the user which in turn increases the payoff of the ASP.

E. Evaluation of the Admission Control Policy

Now, we evaluate the proposed trunk reservation based admission control policy against the simple first come first serve

No.	C_i (Kbps) Peak	C_i (Kbps) TTPP	P^b Peak	P^b TTPP
1	60	40	0.02	0.028
2	440	300	0.045	0.048
3	60	20	0.036	0.038
4	360	260	0.049	0.054
5	240	120	0.044	0.049

TABLE II
COMPARISON OF PEAK PROVISIONING AND TTPP

No.	Synch. η_i^C	Shift η_i^C	Synch. P^b	Shift P^b
1	2.91	2.88	0.028	0.010
2	1.35	1.25	0.048	0.023
3	5.02	4.71	0.038	0.013
4	1.3	1.21	0.054	0.030
5	1.55	1.51	0.049	0.027
ASP	1.06	1.13	–	–

TABLE III
COMPARISON OF TTPP UNDER SYNCHRONIZED AND SHIFTED LOADS

(FCFS) policy. In FCFS, a request to increase capacity allocation is always accepted as long as the service provider has the required capacity, irrespective of the pricing parameters.

An admission control scheme should be tested in situations of overload and unbalance. We increased the arrival rate of user 5 nine fold and performed the simulations, without changing the provisioned capacity. For this case, we also reduced its revenue factor by half. The results are shown in Table IV. Note that the blocking probability of the misbehaving user (user 5) is higher with the trunk reservation based admission control as compared to with the FCFS based admission control. For the well-behaved users, the blocking probability with trunk reservation based admission control is smaller as compared to that with FCFS admission control. Therefore, in the trunk reservation based scheme, well-behaved users are given a protection against the misbehaving user, while in the FCFS scheme there is no such protection and the misbehaving user takes up capacity of the other users. We also found that the revenue of ASP in trunk reservation based scheme is 1.4 times that in the FCFS scheme.

VII. Conclusions

In this paper we proposed a new SLA based on a three tier pricing model for QoS provisioning and dynamic capacity allocation. Our SLA evolves the current paradigm of fixed price and fixed capacity to a situation where users still book a fixed capacity at a fixed price but are allowed to dynamically relinquish parts of their capacities or buy additional capacities based

User No.	P^b FCFS	P^b TR
1	0.252	0.179
2	0.274	0.098
3	0.364	0.265
4	0.257	0.165
5	0.370	0.468

TABLE IV
COMPARISON OF FCFS AND TR

on their needs. Users not interested in the complexity of such dynamic allocation can continue to operate at the fixed capacity model. Some other users can choose a simple time of day based dynamic capacity allocation policy. The most advanced users can implement more sophisticated resizing techniques to get the full benefit of dynamic capacity allocation.

We show how the admission control problem for a service provider using the proposed TTPP SLA can be solved satisfactorily using a trunk reservation based admission control algorithm. This algorithm has low overheads.

We illustrate two scenarios where our proposed SLA can be usefully deployed: application service providers (ASPs) and virtual private networks (VPNs). Using traces of web traffic, we show that the proposed SLA and the admission control policy gives significant gains of the order of 50 to 100 percent in the simulated scenarios.

The proposed SLA may also find applications in other domains where a resource is shared among multiple users and the resource provider charges the users for the use of its resources.

Selecting the parameters of the proposed SLA (revenues, penalties, discounts, and premiums) is an area for future study. Also one needs to study the desired nature of penalties: fixed, delay dependent or proportional. Similarly, from a user viewpoint finding the optimal long term committed bandwidth, and resize strategy is another unexplored area. Finally the proposed trunk reservation approach may not suffice in all scenarios and alternative admission control policies are also of interest.

VIII. ACKNOWLEDGEMENTS

We are highly grateful to Dr. Sandeep Juneja of Indian Institute of Technology Delhi for his guidance. We thank Supratim Deb for excellent discussions related to this work.

REFERENCES

[1] J. Almeida, M. Dabu, A. Manikutty, and P. Cao. Providing differentiated levels of service in web content hosting. Technical Report CS-TR-1998-1364, Computer Sciences Department, University of Wisconsin-Madison, 1998.
[2] User network interface specifications 3.1. ATM Forum, Prentice Hall, 1993.
[3] Private network to network interface specification 1.0. ATM Forum, Prentice Hall, Mar. 1996.
[4] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan. RSVP-TE: Extensions to RSVP for LSP tunnels, Dec. 2001. RFC 3209.
[5] N. G. Bean, R. J. Gibbens, and S. Zachary. Asymptotic analysis of single resource loss systems in heavy traffic, with applications to integrated networks. *Advances in Applied Probability*, 27:273–292, Mar. 1995.
[6] D. D. Clark. A model for cost allocation and pricing in the Internet. In L. McKnight and J. Bailey. Internet Economics. 1^{st} Edition. MIT Press, Cambridge.
[7] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe. A flexible model for resource management in virtual private networks. In *Proceedings of SIGCOMM*, Aug. 1999.
[8] F. Engel. The role of service level agreements in the internet service provider industry. *International Journal of Network Management*, 9:299–301, 1999.
[9] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, Apr. 1990.
[10] S. Floyd and V. Jacobson. Link sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4), August 1995.
[11] R. Garg and H. Saran. Fair bandwidth sharing among virtual networks: A capacity resizing approach. In *Proceedings of INFOCOM*, Tel Aviv, Israel, Mar. 2000.
[12] R. Gibbens and F. Kelly. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information*, 7:77–111, 1990.
[13] R. Gibbens and F. Kelly. Network programming methods for loss networks. *IEEE Journal on Selected Areas in Communications*, 13:1189–1198, 1995.
[14] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A framework for IP based virtual private networks, Feb. 2000. RFC 2764.
[15] P. Goyal, A. Greenberg, C. Kalmanek, B. Marshal, P. P. Mishra, D. Nortz, and K. K. Ramakrishnan. Integration of call signaling and resource management for IP telephony. *IEEE Network*, 13(3):24–32, May-June 1999.
[16] Internet traffic archive. http://ita.ee.lbl.gov/html/traces.html.
[17] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding PHB, June 1999. RFC 2598.
[18] B. Jamoussi and et al. Constraint-based LSP setup using LDP, Jan. 2002. RFC 3212.
[19] F. Kelly. Modelling communication networks, present and future. *Philosophical Transactions of the Royal Society*, A354:437–463, 1996.
[20] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.
[21] P. Key. Service differentiation: Congestion pricing, brokers and bandwidth futures. In *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Basking Ridge, New Jersey, June 1999.
[22] A. Lazar and N. Semret. Design and analysis of the progressive second price auction for network bandwidth sharing. *Telecommunication Systems: Special issue on Network Economics*, 1999.
[23] S. Low. *Traffic Control in ATM Networks*. PhD dissertation, University of California at Berkeley, May 1992.
[24] K. MacKie-Mason and H. R. Varian. Pricing congestible network resources. *IEEE Journal of Selected Areas in Communications*, 13(7):1141–1149, September 1995.
[25] H. Mendelson. Pricing computer services: Queueing effects. *Communications of ACM*, 28(3):312–321, March 1985.
[26] D. Mitra, R. J. Gibbens, and B. D. Huang. State-dependent routing on symmetric loss networks with trunk reservation - I. *IEEE Transactions on Communications*, 41(2):400–411, Feb. 1993.
[27] C. Nicoll. A closer look at service level metrics. *International Journal of Network Management*, 9:287–290, 1999.
[28] R. Pandey, J. F. Barnes, and R. Ollsson. Supporting quality of service in HTTP servers. In *Symposium on Principles of Distributed Computing*, pages 247–256, 1998.
[29] E. Rosen, R. Callon, and A. Viswanathan. Multiprotocol label switching architecture, Jan. 2001. RFC 3031.
[30] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of IEEE*, 83(10):1374–1396, Oct. 1995.
[31] L. Zhang, S. Deering, D. Estrin, S. Shenker, et al. RSVP: A new resource reservation protocol. *IEEE Network*, 7(5):8–18, September 1993.