Maximizing Profit Using SLA-Aware Provisioning

Ananya Das
Math & Computer Science Department
Lake Forest College
Lake Forest, IL 60045
Email: adas@lakeforest.edu

***Abstract*—A Service Level Agreement (SLA) is a contract between a Service Provider (SP) and a customer that typically includes the customer requirements that the SP guarantees, the fee paid to the SP if the requirements are satisfied, and the penalty incurred by the SP if they are violated. Since an important requirement is the customer's service availability, we focus on routing and admission control in optical networks to improve the SP's ability to meet customers' availability requirements.**

Previous researchers used statistical path availabilities to satisfy SLA requirements. A more accurate measure is the actual *probability* that the request will satisfy the SLA requirements. Furthermore, since typically the SP's goal is to maximize profit, a good admission control policy should also consider the profitability of the request.

We study the problem of provisioning connection requests to maximize profit in optical networks. We propose a two-step solution to this problem: first, efficient SLA-aware routing and second, intelligent admission control. For the SLA-aware routing, we consider both single path and pair of paths (one primary and one backup) solutions that route the request while minimizing the SLA violation probability. For the admission control, we propose a model to express the profitability of a request and an admission control policy that considers the violation probability and profitability to determine if and how the request should be admitted. Our admission control policy assesses a request's profitability by considering not only its expected profit but also by quantifying its resource utilization. Our results show that our solution provisions more requests, satisfies more SLA requirements, and yields more expected profit than the traditional approach.

I. Introduction

A Service Level Agreement (SLA) defines performance guarantees made by a network service provider (SP) to its customer. It typically includes the customer's requirements that the SP guarantees to provide, the fee paid to the SP if the requirements are satisfied, the penalty incurred by the SP (usually in the form of a rebate to the customer) if they are violated, and the length of time the agreement holds (referred to as the *penalty period*). An important Quality of Service (QoS) guarantee typically included in the SLA is the customer's service availability. Since large companies often outsource their IT infrastructure to third party SPs, the implications of not meeting SLA guarantees can be serious: a disruption in service can result in significant revenue loss to both the customer and provider. However, since realistically, it may be difficult (and expensive) to satisfy every customer's request, typically, the SP's goal is to provision requests such that profit is maximized.

Previous researchers have proposed availability-based approaches to meeting SLA requirements [1, 2, 9, 15]. Specifically, if a path with availability higher than the availability requirement is found, the path is selected to service the request. However, since availability is a statistical average, this approach is not precise: although a link's failure probability is modeled as uniform over time, failures actually occur at random instants. Rather than statistical availabilities, the admission decision should consider the *probability* that a request will satisfy its SLA requirements. Furthermore, the availability-based approach fails to consider *all* of the factors defined in the SLA. Since typically, the SP's goal is to maximize profit, it is important to consider not only availability satisfaction, but also the economics of a request and how the current request may impact the network's ability to satisfy future requests. For example, requests that are likely to yield high profit without consuming excessive valuable resources should be admitted over requests that yield low profit compared to their resource consumption. Therefore, the SP should use an intelligent admission control policy that judiciously determines which requests are likely to be most profitable. An SP may find that in a highly congested network where resources are scarce, rejecting a few requests may allow better future requests to be admitted, ultimately yielding higher profit.

Clearly, the expected profitability of a request depends on direct costs, specifically the SLA fee and penalty, and the probability that the request meets the availability requirement during the penalty period. However, profitability also depends on indirect costs incurred by the network for admitting the request. This cost, which we refer to as *opportunity cost*, should reflect both the cost of the resources used to route the current request, and how these resources will affect the admission decision of future requests. The direct costs (ie. fee and penalty) are in units of dollars whereas the opportunity costs are in terms of network resources. Therefore, combining these costs requires modeling the opportunity costs in terms of dollars. In an online setting where future requests are not known in advance and several factors (eg. load, link demand/congestion, bandwidth usage) affect the network, determining an exact model for the opportunity cost is a challenging problem. Therefore, our solution is to develop a model that estimates the opportunity cost of a request by quantifying the resource utilization. We then convert this

This work has been supported by NSF Grant No. CNS-05-20190.

978-1-4673-0269-2/12/$31.00 © 2012 IEEE

resource utilization to an opportunity cost (in dollar terms) by considering how much profit we typically get as resources are used. Our simulation results show that our model effectively estimates the opportunity cost. To our knowledge, this is the first work to perform admission control by estimating a requests' profitability based on both its expected profit and the long-term effects of its resource utilization.

Our overall goal is to efficiently route (or reject) SLA requests to maximize overall profit. We propose a two-step solution to this problem: first, efficient SLA-aware routing, and second, intelligent admission control. Since single path solutions may not always be reliable enough to meet SLA requirements, we look for two routing solutions: (1) a single path and (2) a pair of paths (pair-paths), that route the request while minimizing the SLA violation probability. For the admission control, we evaluate the profitability of a request to determine whether it should be admitted. To do this, we first propose a model that quantifies the resource utilization of a request to effectively express its opportunity cost. We then propose an admission control policy that considers the opportunity cost along with violation probability of both routing solutions (single path and pair-paths) to determine if the request should be admitted, and if so, which solution will be more profitable. Our simulation results show that our algorithm provisions more requests, satisfies more SLA requirements, and yields more expected profit than the traditional availability-based approach.

Our approach allows SPs to adjust parameters based on their customer's specific needs. For example, SPs who service hospitals or other institutions with critical needs may need to always provide highly reliable service, whereas SPs who provide entertainment services may be able to provide less reliable service without losing revenue. To investigate how our admission control policy performs for various parameters, we compare three variations of the policy: (1) *Admit-All*, where all requests that can meet their bandwidth requirement are admitted regardless of their violation probabilities, (2) *Admit-Most*, where the requests that are likely to be most profitable are admitted, and (3) *Admit-Few*, where only requests with the highest SLA satisfaction probabilities are admitted. We found that for our settings (which are similar to those used in previous studies such as [6, 8, 10, 12]) the *Admit-Most* policy was most beneficial as it yielded both high expected profit and low violation rate.

We recognize that some SPs prefer to accept all requests (that meet their bandwidth requirement) regardless of their SLA requirements and therefore do not require an admission control strategy. These SPs can use our opportunity cost model to determine an effective pricing policy. For example, requests with high opportunity cost use more resources and therefore should be assigned higher fees.

Our main contributions are as follows:

- We propose a new SLA-based approach to routing and admission control. Unlike previous approaches that ignore many aspects of the SLA, our algorithm uses SLA specifics to estimate a request's satisfaction probability, expected profit, and impact on future requests.
- We develop a model to quantify the resource utilization of a request. This value, which we refer to as the *opportunity cost*, measures how admitting the request will affect the network's ability to admit future requests.
- We propose an efficient routing approach that finds paths with low violation probability. Our routing algorithm also reduces link congestion.
- We compare our algorithm to a traditional availability-based approach and find that it performs significantly better in terms of admittance, overall satisfaction, satisfaction among admitted requests, and expected profit.

The remainder of this paper is organized as follows. Section II discusses previous SLA-related research. Section III describes the SLA-provisioning problem we are considering. Section IV describes our provisioning algorithm and Section V discusses the results of our algorithm. Finally, Section VI provides a summary of our work.

II. Related Work

Many researchers have studied various approaches to satisfying SLAs with different requirements ([4, 5, 7, 11–14]).

The authors of [13] consider the problem of satisfying SLAs with survivability requirements. Their goal is to minimize the number of wavelengths used while maximizing the number of satisfied connections. The authors assume that the SLA of each request indicates whether the request should be satisfied with a single path or two paths where one is the primary path and the other is a link-disjoint back up path (ie. full protection). One drawback to this approach is that the SP does not have full control over resource management. For example, the SP may have to use valuable network resources to satisfy a request that requires full protection but yields low revenue, which may cause more profitable requests to be blocked in the future. Furthermore, the algorithm proposed in [13] assumes that survivability is the only SLA metric, and does not consider availability, bandwidth, or pricing.

In [7], the authors estimate SLA violation probabilities based on SLA requirements, link availabilities, and failure repair times. They find that SLA violation is greatly influenced by the availability requirement and the repair time. The authors of [12] present an algorithm for reducing SLA violation probability using shared-path protection. The authors of [14] also consider satisfying availability requirements with shared-path protection. They use a Markov model to estimate the SLA violation probability based on the required availability, link availability, failure repair times, and the penalty period. However, they do not consider the effect of bandwidth requirements. In [4], the authors consider satisfying SLA requirements using multipath provisioning while minimizing jitter. Although each of these studies examines variations of SLA satisfaction, none consider the economics of SLAs.

The authors of [3] use Integer Programming to propose a model for satisfying path availabilities while minimizing an operational cost that is based on the number of lightpaths used to satisfy the request. However, their model attempts to only maximize the path availability and makes no QoS *guarantees*. The authors of [11] propose an algorithm for minimizing SLA violations while maximizing profit. They employ reprovisioning, where high-priority connections may preempt backup resources from low-priority resources. Although they consider the revenues and fees of a request, they do not consider future cost implications of satisfying the request (ie. opportunity cost).

The authors of [5] present a profit-analysis model and compare SLAs with varying protection requirements. They find that providing mixed service (ie. full protection for some requests while no protection for others) is most profitable.

Our work focuses on satisfying availability requirements of SLAs while achieving high profit. The authors of [1, 2, 9, 15] proposed satisfying availability requirements by considering only path availabilities: if a path with availability higher than the requirement is found, the request is accepted. However, since path availabilities are statistical averages, they do not reflect the randomness of network failures, and therefore may not provide a complete measure of path reliability. Therefore, we base the admission decision on the actual *probability* that the request will satisfy the SLA requirements. Since our goal is to minimize SLA violations while maximizing profit, we also consider the profitability of the request.

III. SLA-Based Provisioning

A. Problem Statement

In the SLA-based provisioning problem that we study, bandwidth requests arrive dynamically and each request must be scheduled (or rejected) as it arrives. Once a request has been scheduled, it cannot be rerouted. The network is represented by a directed graph $G = (V, E)$, where V is a set of vertices and E is a set of edges such that each edge in E has an availability in (0, 1) and a non-negative integer capacity. Each connection request is associated with a SLA that declares the source, destination, bandwidth requirement, penalty period (the length of time the agreement holds), and availability requirement. The SLA also declares the fee obtained by the SP for meeting the availability requirement and the penalty incurred by the SP for violation. Specifically, connection requests are of the form $<s, d, b, a, T, f, y>$, where $s, d \in V$, s is the source, d is the destination, b is the bandwidth requirement, and a is the availability requirement.

At the end of the penalty period, T, the customer and the SP determine if the customer owes a fee or if the SP must pay a penalty. The availability requirement can be expressed in terms of an allowed downtime (ADT) and the penalty period. Specifically, the ADT is $(1 - a) \times T$.

The fee, f, is paid by the customer if the SP provides a connection that satisfies the availability requirement during the penalty period; otherwise, the penalty y is paid by the SP if it is unable to meet the requirement. For each request, the SP must decide whether to admit the request, and if so, how to route it.

Our solution has two major components: first, we use an effective SLA-aware routing algorithm to find good paths with low violation probability; second, we use an intelligent admission control policy that admits requests based on their expected profit and an estimate of the value of the network resources they use.

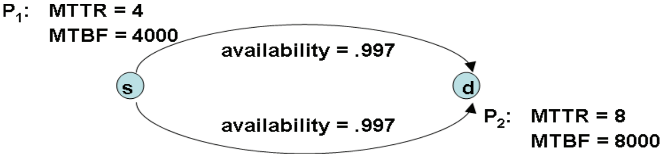

Fig. 1. Although paths P_1 and P_2 have equal availabilities, they have unequal MTTRs and MTBFs, so their SLA satisfaction probabilities are unequal.

B. SLA-Based versus Availability-Based Provisioning

Previous researchers ([1, 2, 9, 15]) proposed methods of satisfying SLA requirements based on path availability: if the overall statistical availability of a path is greater than the availability requirement of the SLA, the path may be selected to service the request. Since availability, a statistical average, may not accurately describe a link's failure probability, our algorithm chooses paths based on their estimated SLA violation probability.

Formally, we define an SLA violation as the event that the actual downtime of a request exceeds the allowed downtime (ADT). We assume that link failures and failure repair times follow known distributions, and that failures occur independently (an interesting extension of this work would consider shared risk link groups). As described above, for a given request r, if a_r is the required availability and T_r is the penalty period, then the ADT is $(1 - a_r) \times T_r$. For a single path, the actual downtime is the sum of the downtimes from each failure that occurs on the path. For a pair of paths, where one path is the primary path, and the other is a backup in case the primary fails, the downtime is the total time that the paths are down *simultaneously*.

Figure 1 provides an example that illustrates why path availability may not be an accurate measure of SLA satisfaction probability. In the example, suppose a request r is issued from s to d with $T_r = 30$ days and ADT of 3 hours ($a_r \approx 0.996$). Let P_1 and P_2 denote two paths with equal availabilities of 0.997, and exponentially distributed mean times to repair (MTTR) of 4 hours and 8 hours, respectively. Since the MTTRs for P_1 and P_2 are unequal, the mean times between failures (MTBF) will also be unequal, so P_1 and P_2 will have unequal SLA satisfaction probabilities. Specifically, the satisfaction probabilities for P_1 and P_2 are 0.803 and 0.837, respectively, so P_2 is the better choice.

IV. Our SLA-Based Provisioning Algorithm

The goal of our provisioning algorithm is to maximize the Service Provider's overall profit. Our algorithm consists of two main steps: (1) efficient routing and (2) intelligent admission control. For our problem, we assume that network resources are fixed and known in advance. We focus on profits from fees for accepted requests whose SLA is satisfied, but we do not consider the capital costs for building nor the operational costs of maintaining the network.

For each request, our algorithm decides whether to admit the request and if so, how to route it. Our routing algorithm reduces SLA violation probability and uses resources efficiently to preserve resources for future requests (we describe this in more detail in Sections IV-A1 and IV-A2).

Our novel contribution is our admission control policy which assesses the profitability of a request by considering not only its expected profit but also its *opportunity cost*. The expected profit is based on the fees and penalties specified in the SLA and the probability that the SLA requirements will be satisfied. The opportunity cost is a measure of the cost of the resources used to satisfy the request (we describe this in more detail in Section IV-B1). Our admission control strategy considers a request's profit and its opportunity cost to assess its overall value (we describe this in more detail in Section IV-B2): if the profit is high compared to the opportunity cost, the request is considered valuable.

A. Routing Algorithm

1) SLA-Based Routing: Our routing strategy aims to find paths that minimize SLA violation probability and maximize profit. Given a connection request, our routing algorithm searches for two solutions: (1) a single path with minimum violation probability and (2) a pair of paths (where one path is the primary path and the other is a backup in case the primary fails) with minimum joint violation probability. It then chooses the solution that maximizes the expected profit (we describe expected profit in Section IV-B).

We first describe how our algorithm estimates the violation probabilities for a single path and a pair of paths. Let r denote a connection request with ADT ADT_r, and let P denote a single path used to route r. If DT_P is the actual downtime on P, then we denote the probability that using P to route request r violates the SLA as $Pr(DT_P > ADT_r)$. If r is routed with two paths P_1 and P_2 such that P_1 is the primary path, and P_2 is used as a backup path in case P_1 fails, then the SLA is violated if the time that P_1 and P_2 are *simultaneously* down exceeds the ADT . If DT_{P_1,P_2} denotes the total simultaneous downtime of P_1 and P_2, then we denote the probability that P_1 and P_2 violate r as $Pr(DT_{P_1,P_2} > ADT_r)$. To estimate the violation probabilities, our algorithm assumes that all failures (ie. multiple failures on the same path, and failures on multiple paths) arrive independently. Our algorithm assumes that failure repair times follow a specific distribution. We focus on the exponential distribution, but also test our algorithm when it assumes other distributions for failure repair times. (Derivations for SLA violation probabilities have been omitted due to space constraints.)

We now describe how our algorithm finds the candidate solutions. Our routing algorithm starts by assigning each link $\ell \in E$, a weight w_ℓ (initially set to 1). To find the single path solution, our algorithm creates a large set of low-weight paths between the source and destination and chooses the path that has enough free capacity to meet the bandwidth requirement and minimum violation probability. For the pair-paths solution, we use the technique proposed in [8] to create a large set of pairs of low-weight *disjoint* paths between the source and destination. The basic idea of this technique is to first find a set S containing low-weight disjoint paths between the source s and destination d. Then by splicing together subpaths of paths from S, we create a large collection of low-weight $s-d$ paths. From this collection of paths, we extract pairs of disjoint paths. From this set, we choose the pair with enough free capacity on each path and minimum joint violation probability.

Once the algorithm determines the candidate solutions, it computes the opportunity cost and expected profit of each (described below) and selects the solution that maximizes the expected profit.

2) Reducing Congestion: In addition to minimizing violation probability, our routing algorithm also aims to reduce link congestion. When a link ℓ, is used, we increment its weight, w_ℓ. Furthermore, when a request departs a link, we decrement its weight. Therefore, congested links will have high weights and will therefore be less likely to be chosen for a future request. This technique is easy to implement and (based on simulations) significantly improves the performance of our algorithm in terms of both satisfaction probability and profit.

B. Admission Control

1) A Model for Opportunity Cost: The goal for our admission control is to admit the most profitable requests. A request's profitability is based on direct costs (ie. its fee and penalty) and its opportunity cost. The opportunity cost should reflect how admitting the request will affect our ability to admit future requests. Determining an exact model for the opportunity cost is difficult since the setting is online and many factors are involved (eg. load, link demand/congestion, bandwidth usage). Therefore, our approach is to estimate the opportunity cost by quantifying the resource utilization of the request. Using resources to satisfy the current request may either (1) prevent us from satisfying certain requests in the future, or (2) force us to use less reliable links to satisfy future requests, causing these connections to be less reliable. In either case, although the use of costly resources may yield a higher profit for the current request, it can result in lower profits for future requests.

Our opportunity cost model reflects both the marginal costs of the request and the future cost implications. The marginal cost is determined by the network resources consumed (ie. bandwidth). The future cost is indicated by the relative value of these resources. For example, high-demand resources (ie. links that are frequently used) are considered more costly than low-demand ones. Since the weight w_ℓ of link ℓ (described in Section IV-A1) reflects the number of times ℓ has been used, it measures the demand for link ℓ, so requests that have a high sum of link weights are considered costly. Similarly, requests that require high-availability links are considered more costly than those that can be satisfied with less reliable links.

Let $\bar{P}$ denote the path or pair of paths found to route request r. We define the opportunity cost, C_r of r as follows:

$$C_r = b \cdot \frac{1}{1 - a_{\bar{P}}} \cdot \sum_{\forall \ell \in \bar{P}} w_\ell \tag{1}$$

where b is the bandwidth requirement, $a_{\bar{P}}$ is the availability[1] of $\bar{P}$, and w_ℓ is the weight of link ℓ, or the number of times the link has been used to route a request.

[1] The joint availability of two paths with availabilities a_1 and a_2 is $a_1 + (1 - a_1)a_2$.

Notice that this model for the opportunity cost reflects both the marginal costs of the request and the future cost implications. The marginal cost is indicated by the bandwidth consumption. The future cost is indicated by the relative value of the resources consumed: requests that use high-demand links or highly available paths will have higher opportunity costs.

SPs who prefer to admit all requests regardless of their SLA requirements can use this opportunity cost model to help determine an effective pricing policy for requests. More specifically, requests with higher opportunity costs should be assigned higher fees.

2) Profit-Based Admission Policy: The goal of our admission control policy is to admit requests with high expected profit compared to their resource usage. Our admission control policy determines the profitability of a request based on its opportunity cost and its expected profit. Again, let $\bar{P}$ denote the path (or pair of paths) found to route a request r. Let $DT_{\bar{P}}$ denote the experienced downtime on $\bar{P}$ and let ADT_r denote the ADT for r. Formally, *the expected profit*, E_r of request r is:

$$E_r = f_r - Pr(DT_{\bar{P}} > ADT_r) \cdot y_r \tag{2}$$

where f_r is the fee paid to the SP if request r is satisfied and y_r is the penalty paid by the SP if r is violated.

Our admission control policy works as follows. We first determine the expected profit for one unit of opportunity cost for a typical request. This value, which we refer to as the *global normalized profit* (GNP), measures how much expected profit a typical request *should* yield in return for its resource usage. To compute the GNP, we perform a preliminary run of our algorithm on a typical set of requests where we admit all of them. For each request, we compute the ratio of expected profit to opportunity cost (ie. $\frac{E_r}{C_r}$); the GNP is the average of these ratios. As we describe below, we determine the relative worth of a connection request by comparing its profitability to the GNP.

Specifically, when a connection request is received, we first find 2 candidate routing solutions: a single path P, and a pair of paths P_1 and P_2 (see Section IV-A1 for more details). For each solution, we compute the expected profit per unit of opportunity cost, (again $\frac{E_r}{C_r}$) that would be obtained by using the solution to route r. We refer to this value as the *local normalized profit* (LNP). Typically, requests will have relatively high opportunity costs if they use more reliable edges, more popular edges, or more bandwidth (or a combination of these), and should therefore yield higher expected profits. These requests will therefore have high LNPs. If either routing solution yields an LNP that is close to the GNP, then the request is likely to be profitable, and is therefore admitted. Specifically, let Γ denote the GNP, Λ_P denote the LNP for the single path solution, and Λ_{P_1,P_2} denote the LNP for the pair-paths solution. Then for some threshold value δ, if $\Gamma - \delta \leq \Lambda_P$ or $\Gamma - \delta \leq \Lambda_{P_1,P_2}$, then r is admitted; otherwise r is rejected. If r is admitted, we choose the routing solution which yields a higher LNP. Figure 2 illustrates our admission policy.

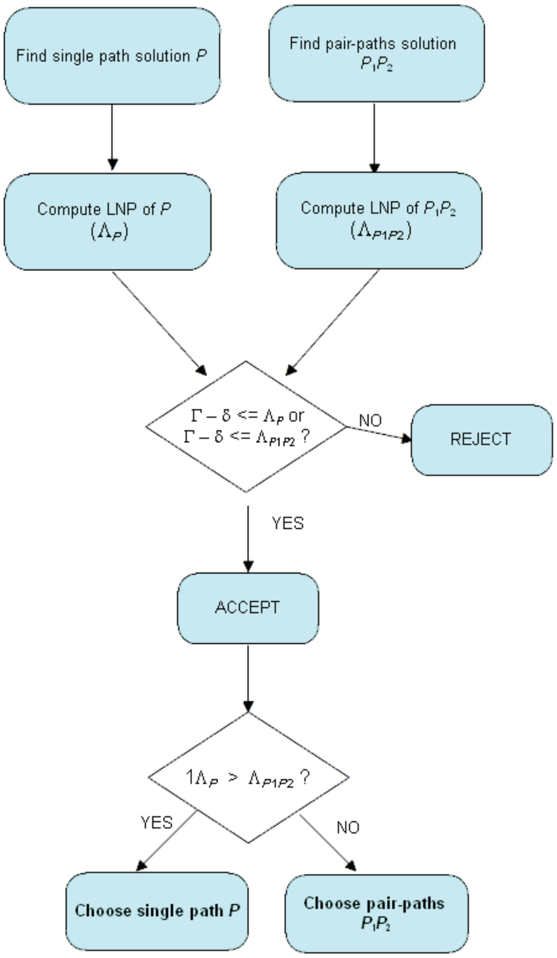
Find single path solution P
Find pair-paths solution P_1P_2
Compute LNP of P (Λ_P)
Compute LNP of P_1P_2 (Λ_{P1P2})
$\Gamma - \delta <= \Lambda_P$ or $\Gamma - \delta <= \Lambda_{P1P2}$?
NO
REJECT
YES
ACCEPT
$1\Lambda_P > \Lambda_{P1P2}$?
YES
NO
Choose single path P
Choose pair-paths P_1P_2

Fig. 2. Admission Control Policy. Γ is the pre-computed Global Normalized Profit and δ is a threshold value.

We tested various values of δ and for our simulations, we chose a value that yielded both high profit and low violation probability. However, SPs can set this threshold based on their service goals (eg. a large δ may increase the number of admitted requests, a small one may lower the fraction of violated requests, and an intermediate one may maximize the total number of satisfied requests). Algorithm 1 describes our provisioning algorithm in detail.

V. Numerical Results

To evaluate the performance of our algorithms, we simulated a dynamic network environment similar to those used in previous studies ([6, 8, 10, 12]). The request arrival process is Poisson and the request holding-time follows a negative exponential distribution with unit mean. For simplicity, we assume failures occur independently. For the results we present, we assume that our algorithm knows the distribution of failure repair times. We tested our algorithm using a variety of distributions: exponential (exp(1/MTTR)),

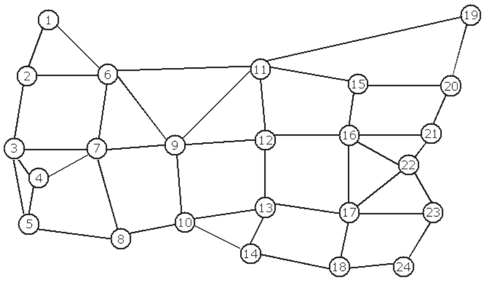

Fig. 3. Sample network topology.

normal ($\text{N}(\text{MTTR}, (.3\times\text{MTTR})^2)$ and $\text{N}(\text{MTTR}, (.8\times\text{MTTR})^2)$), and uniform ($\text{U}(.7\times\text{MTTR}, 1.3\times\text{MTTR})$ and $\text{U}(.2\times\text{MTTR}, 1.8\times\text{MTTR})$). Although we present only the results of the exponential distribution, we found that for all distributions, our algorithm outperformed the previous approach in terms of admission, SLA satisfaction, and expected profit. We also tested our algorithm under a setting where the algorithm's assumed repair time distribution does not match the actual one, and found that although it does not perform as well, it still outperforms the previous approach. This shows that more accurate knowledge of the distribution of repair times is helpful; this information may be obtained using statistical data on the actual repair times.

For our simulations, we used the sample topology shown in Figure 3. The network is fully wavelength convertible. There are 16 wavelengths per link, and the capacity of each is OC-192 (≈10Gbps), a realistic measure for today's channel speeds. The bandwidth requested is an integral multiple of STS-1 (≈50Mbps). Link availabilities are uniformly distributed over {.99, .999, .9999}.

SLA requirements are distributed as follows. Requested availabilities are uniformly distributed over four service classes: .995, .999, .9999, .99999, and penalty periods are uniformly distributed over either 1 or 2 months. Bandwidth requirements are distributed such that 85% of the requests are for OC-1, 10% are for OC-12, and the remaining 5% are for OC-96. These settings are similar to those used in previous studies ([8, 12]). SLA fees and penalties are linearly correlated with the bandwidth and availability requirements. Specifically, if b is the bandwidth request, then the SLA fee is $50b$, $100b$, $200b$, or $300b$ for availability requirements of .995, .999, .9999, and .99999, respectively. For every request, the fee is equal to the penalty, so if the requested availability is not satisfied, the fee must be forfeited. (We also tested other pricing policies that are described in Section V-A.) We simulated 10,000 connection requests for various load levels[2].

A. Comparing to Availability-Based Provisioning

We first compare our SLA-based provisioning algorithm to the availability-based approach. Figure 4 shows the admittance rate for each algorithm. For low load (10-30 Erlangs), our SLA-based algorithm admits approximately 1.5 times as many

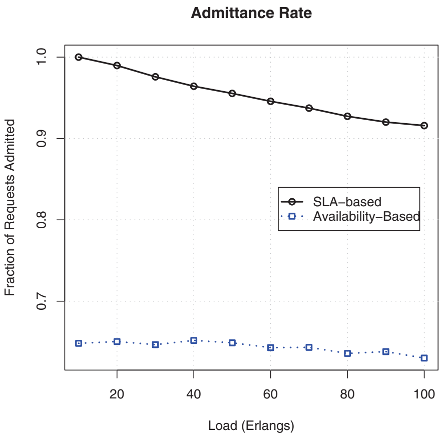

Fig. 4. Admittance Rates for SLA-Based Provisioning versus Availability-Based Provisioning

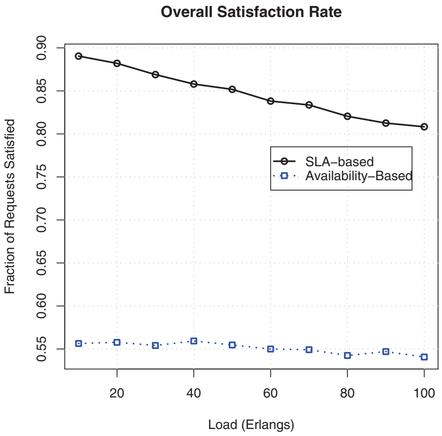

Fig. 5. Fraction of Satisfied Requests (out of all requests) for SLA-Based Provisioning versus Availability-Based Provisioning

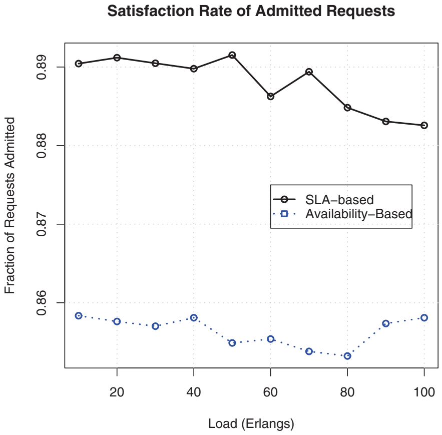

Fig. 6. Fraction of Satisfied Requests (out of admitted requests) for SLA-Based Provisioning versus Availability-Based Provisioning

[2] Load, measured in Erlangs, is defined as the product of the connection arrival rate, the average connection-holding time, and a connection's average bandwidth normalized to units of OC-192.

Algorithm 1. Given: Graph $G = (V, E)$,
$\Gamma =$ *Global Normalized Profit (GNP)*, δ is admission threshold

```
Assign each link $\ell \in E$ a unit weight $w_\ell$.
for all requests $< s, d, b, a, T, f, y >$:
Find a set of low-weight single $s - d$ paths. Choose the path, $P$, with capacity at least $b$ and minimum violation probability.
Find a set of low-weight pairs of disjoint $s - d$ paths. Choose the pair of paths, $P_1$ and $P_2$, such that both paths have capacity at least $b$, and the system of paths has minimum joint violation probability.
$\Lambda_P \leftarrow$ local normalized profit (LNP) for single path $P$
$\Lambda_{P_1,P_2} \leftarrow$ LNP for the pair of paths $P_1$ and $P_2$
if $\Gamma - \delta > \Lambda_P$ and $\Gamma - \delta > \Lambda_{P_1,P_2}$, then
    Reject the request.
else if $\Lambda_P \geq \Lambda_{P_1,P_2}$ then
    Route the request using $P$. $\bar{P} \leftarrow P$.
else
    Route the request using $P_1$ and $P_2$. $\bar{P} \leftarrow P_1 \cup P_2$.
For each link $\ell$ in $\bar{P}$, reduce the capacity of $\ell$ by its new flow and increment $w_\ell$.
When a request departs:
Decrement the weight of all links used to route the request and restore capacity on these links.
```

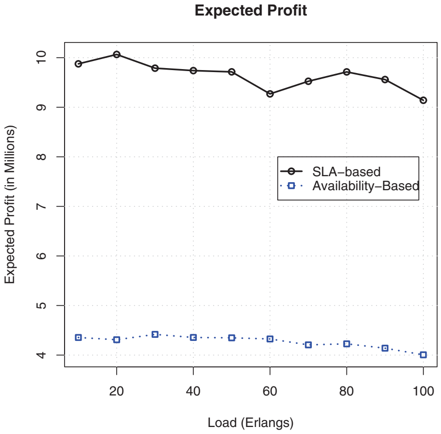

Fig. 7. Expected Profit (in millions) for SLA-Based Provisioning versus Availability-Based Provisioning

requests as the availability-based approach. Similarly, Figure 5 shows that our algorithm satisfies at least 1.5 times as many requests as the competing algorithm. Figure 6 shows that though our algorithm admits significantly more requests than the competing algorithm, it is still able to satisfy a higher percentage of the availability requirements of the requests that it admits. The expected profits acquired by each algorithm are shown in Figure 7. Our algorithm achieves significantly more profit than the previous algorithm (more than twice as much for all load levels). We note that since the expected profit heavily relies on the fees and penalties, the comparison shown is dependent on the pricing policy and may vary somewhat in real-world settings. The results shown in Figs. 4-6 provide a more realistic comparison since these results do not heavily rely on predefined values. However, to get an understanding of how our algorithm performs under varying pricing policies, we also tested the algorithms using different pricing schemes (specifically, for fee f and penalty y we use: $f = 2 \times y$ and $f = 3 \times y$) and found that our algorithm still outperformed the previous approach.

B. Comparing Admission Control Strategies

An important advantage of our admission control policy is that it can be easily adapted to fit the needs of SPs with different goals. By adjusting the admission threshold, δ (see Section IV-B2), the SP can control the stringency of the policy. We now compare three adaptations of our admission control policy: (1)*Admit-All* (2)*Admit-Most*, and (3)*Admit-Few*.

In the *Admit-All* scheme, all requests are admitted, regardless of their expected profits and violation probabilities, as long as there is sufficient network capacity. An SP whose goal is to accommodate as many customers as possible, at the risk of violating several SLAs, may choose this policy.

In the *Admit-Most* scheme, requests that are likely to be profitable are admitted and less profitable requests are rejected. An SP whose goal is to accommodate many customers while achieving high customer satisfaction by providing typically reliable service, may choose this moderately aggressive admission policy. *Admit-Most* is the approach we took for our simulation results in Section V-A. For our simulations, we used a fixed threshold value and admitted the requests that yielded a profit higher than this threshold. A useful feature of this approach is that SPs can tune this threshold according to their preferences.

Finally, in the *Admit-Few* scheme, a request is admitted only if its satisfaction probability is higher than a predefined threshold (as in the Admit-Most scheme, SP's can tune this threshold value). An SP whose goal is to always provide highly reliable service may choose this policy.

The choice for the admission control scheme may also depend on the urgency levels of requests. For example, SPs who service hospitals or other medical institutions may choose the *Admit-Few* scheme to ensure that all of their connections will be very reliable. On the other hand, SPs who provide less urgent services may find that the more flexible *Admit-All* or *Admit-Most* scheme is a better fit for their needs.

Figures 8-9 show that the *Admit-Few* policy achieves the highest rate of satisfaction among admitted requests, although this approach yields the lowest expected profit. On the other hand, the *Admit-All* policy yields the highest expected profit but violates the SLA of many admitted requests. Note that using the *Admit-All* policy may result in another indirect cost, namely, the long-run cost of losing customers (and potential future revenue from them) due to SLA violations. The *Admit-Most* policy achieves both high satisfaction without sacrificing much profit, and therefore may be a good approach for many realistic settings. The results in Figs. 8-9 indicate a notable tradeoff between request satisfaction and profit.

Note that although the *Admit-All* policy may initially yield high profit, this profit may not be long-term since customers are likely to switch to a more reliable SP. Therefore, SPs should decide, based on their goals, whether it would be more

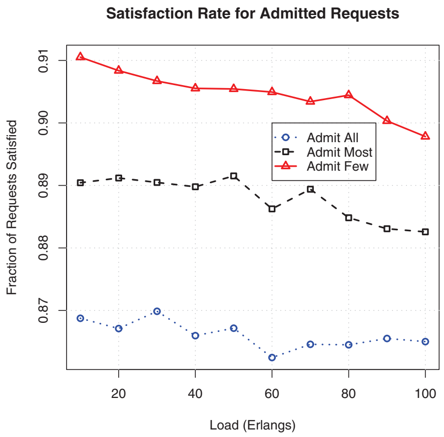

Fig. 8. Fraction of Satisfied Requests (out of admitted requests) for three adaptations of our admission control: *Admit-All*, *Admit-Most* and *Admit-Few*

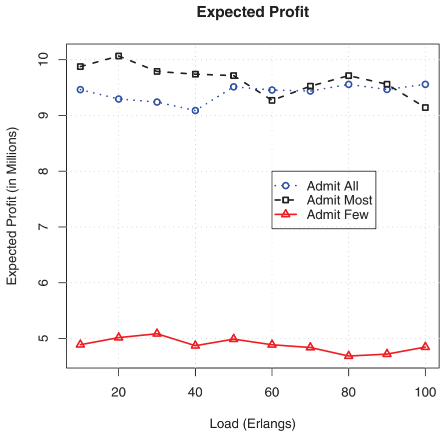

Fig. 9. Expected Profit (in millions) for three adaptations of our admission control: *Admit-All*, *Admit-Most* and *Admit-Few*

beneficial to reject some requests but satisfy a large portion of admitted requests, or admit all requests and violate several of them.

VI. Conclusion

We proposed a profit maximizing solution for provisioning and admission control of SLA requests. Most previous algorithms used statistical availabilities to determine if a request can be provisioned to meet its SLA requirements; however, our algorithm considers the actual SLA satisfaction probability. Our algorithm also quantifies the resource utilization, which we refer to as the *opportunity cost*, of a request to determine how it will affect the networks ability to admit future requests. Our approach performs (1) efficient SLA-aware routing that considers both single paths and pairs of paths to route a request with low violation probability, and (2) intelligent admission control that admits a request based on its expected profit and opportunity cost. Results show that our algorithm outperforms the previous approach in terms of admission, SLA satisfaction, and expected profit.

An advantage of our approach is that it allows SPs to adjust parameters based on their admission goals and the urgency of the requests they receive. We compared three variations of our admission policy: (1) *Admit-All*, where all requests that could meet their bandwidth requirement were admitted regardless of their violation probabilities, (2) *Admit-Most*, where the requests that were likely to be most profitable were admitted, and (3) *Admit-Few*, where only requests with the highest satisfaction probabilities were admitted. Our simulation results show that for our network settings, *Admit-Most* is best as it achieves both high profit and high satisfaction rates for admitted requests.

References

[1] E. Al Sukhni, H. Mouftah. A Novel Distributed Destination Routing-Based Availability-Aware Provisioning Framework for Differentiated Protection Services in Optical Mesh Networks. In *IEEE Symposium on Computers and Communications*. July 2008.

[2] S. Benlarbi. Estimating SLAs Availability/Reliability in Multi-Services IP Networks. *In Lecture Notes in Computer Science*, vol. 4328, pp. 30-42. 2006.

[3] H. Chang, P. Wang, C. Chan, C. Lee. A New Service Level Agreement Model for Best-Effort Traffics in IP over WDM. In *IEEE International Conference on Advanced Networking and Applications*. March 2008.

[4] P. Dey, A. Kundu, M. Naskar, A. Mukherjee, M. Nasipuri. Dynamic Multipath Bandwidth Provisioning with Jitter, Throughput, SLA Constraints in MPLS over WDM Network. In *International Conference on Distributed Computing and Networking*. January 2010.

[5] O. Gerstel and G. Sasaki. A General Framework for Service Availability for Bandwidth-Efficient Connection-Oriented Networks. In *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, June 2010.

[6] D. Griffith, K. Sriram, S. Lee, and N. Golmie. Restorability versus Efficiency in $(1:1)^n$ Protection Schemes for Optical Networks. In *IEEE International Communications Conference*. June 2004.

[7] A. Snow, G. Weckman, V. Gupta. Meeting SLA Availability Guarantees through Engineering Margin. In *International Conference on Networks*. April 2010.

[8] S. Huang, B. Mukherjee, and C. Martel. Survivable Multipath Provisioning with Differential Delay Constraint in Telecom Mesh Networks. In *IEEE Infocom*. April 2008.

[9] R. He, B. Lin, and L. Li. Dynamic Service-Level-Agreement Aware Shared-Path Protection in WDM Mesh Networks. *Journal of Network and Computer Applications*, vol. 30, no. 2, pp. 429-444. 2007.

[10] K. Lu, G. Xiao, and I. Chlamtac. Analysis of Blocking Probability for Distributed Lightpath Establishment in WDM Optical Networks. In *IEEE/ACM Transactions on Networking*, vol. 13, no.1. February 2005.

[11] M. Xia, M. Batayneh, L. Song, C. Martel, and B. Mukherjee. SLA-Aware Provisioning for Revenue Maximization in Telecom Mesh Networks. In *IEEE Globecom*. November 2008.

[12] M. Xia, M. Tornatore, C. Martel, and B. Mukherjee. Risk-Aware Routing for Optical Transport Networks. In *Proceedings of IEEE Infocom*. March 2009.

[13] J. Thangaraj, P. Mankar, R. Datta. Improved Shared Resource Allocation Strategy with SLA for Survivability in WDM Optical Networks. In *Journal of Optics*, vol. 39, no. 2, pp. 57-75. March 2009.

[14] H. Waldman and D. Mello. SLA-Aware Survivability. In *Journal of Networks*, vol. 5, no. 2. February 2010.

[15] J. Zhang, K. Zhu, H. Zang, N. Matloff, and B. Mukherjee. Availability-Aware Provisioning Strategies for Differentiated Protection Services in Wavelength-Convertible WDM Mesh Networks. In *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 11771190. October 2007.