Provisioning End-to-End Quality of Service with Fast Importance Sampling based Traffic Engineering

Benjamin Feng [(1)], Changcheng Huang[(2)], Michael Devetsikiotis[(3)], and Peter Rabinovitch[(4)]

(1)(2)Carleton University, The Department of Systems and Computer Engineering, 1125 Colonel By Drive, Carleton University, Ottawa, Ontario, K1S 5B6, Canada
(1)zmbfeng@sce.carleton.ca, (2) huang@sce.carleton.ca

(3) North Carolina State University, B-104 Flex,Campus Box 7911 – Flex, NC State University, Raleigh, NC 27695-7911,U.S.A, mdevets@eos.ncsu.edu

(4) Alcatel Canada, 600 March Road, P.O. Box 13600, Ottawa, Ontario, K2K 2E6, Canada
peter.rabinovitch@alcatel.com

Abstract: Simulation is a popular tool of network traffic engineering. Knowing how the network performance will change with respect to the network parameters is the key to achieve a target performance (QoS). The traditional crude Monte Carlo (CMC) method [1] is very time consuming, especially for rare event simulation. A previously proposed Importance Sampling based Traffic Engineering (ISTE [2]) approach is faster than the CMC approach yet ISTE may still require additional sample performances aside from the initial sample performance. The Fast Importance Sampling based Traffic Engineering (FISTE) approach can carry out all the tasks of ISTE requiring only a single sample performance thus making FISTE even more time efficient.

Keywords: simulation, traffic engineering, importance sampling, self-similar

1. Introduction

Due to the complexity of many large networks, using analytical or mathematical approaches to predict the network performance, as the network parameters are adjusted, is difficult if not impossible. That is the reason that simulation has become a popular solution to this problem. However, for rare event simulation, the simulation run length will be very long, or many number of independent replication runs will be required. All these will result in longer simulation time.

Imagine under a self-similar [3] [4] traffic model, the target probability of buffer overflow for a network path is set at 10^{-6}. The target probability is to be achieved by shaping the mean rate of the ingress traffic flows at the edge of the network. If the network consists of many nodes and links, each node has different behavior, and each link has different capacity, it is hard to analytically determine how much of the ingress traffic flows (their mean rates) need to be reduce to achieve the target. If the Crude Monte Carlo (CMC) simulation approach is used, the mean rate of the simulated ingress traffic can be reduced at each simulation trial until the simulation reports that the probability of buffer overflow have met the target.

However, as the probability gets closer to the target, each simulation run will require at least 1,000,000 (see section 2.2) independent replications since the target probability is 10^{-6}. The large number of replications makes the length of the entire simulation experiment very long and inefficient. (Note that a self-similar traffic model is used in the simulation, the Batch-Mean [5] method can not be used due to the high correlation of the traffic process; thus, the Replication- Deletion [5] method is used).

2. Background

A previously proposed ISTE [2] approach is based on a variance reduction technique [6]: Importance Sampling [6]. By using known information of the simulation model, the variance reduction techniques can give more accurate estimation of the expectation of some performance measure. Usually, the first stage of a variance reduction technique is to get the simulation model information through an initial crude simulation run; the results from the first stage will be used to improve the accuracy of the performance estimator in the second stage.

2.1. Importance Sampling

Assume $\boldsymbol{Y}$ is the input to the network, $\boldsymbol{Y}_1$,.., $\boldsymbol{Y}_N$ are the random i.i.d.(independent identical distributed) samples from the pdf $f(\boldsymbol{y})$. (note that $\boldsymbol{Y}_i$ is actually a vector/a sample trace, with respect to time, since $\boldsymbol{Y}$ is a random process), and $L(\boldsymbol{Y})$ is some sample performance of the network. The expectation of the sample performance is of our interest: $l=E\{L(Y)\}$. The expectation is:

$$l = E\{L(Y)\} = \int L(y) f(y) dy \tag{1}$$

If we use a CMC estimator for l, then:

$$\bar{\ell}_N = \frac{1}{N}\sum_{i=1}^{N} L(Y_i) \tag{2}$$

In the Replication-Deletion simulation framework, $\boldsymbol{Y}_i$ is the sample input of the i-th independent replication and $L(\boldsymbol{Y}_i)$ is the sample output performance of the i-th replication. For example, if we are interested in the probability of buffer overflow, then:

$$L(Y_i) = \begin{cases} 0, \text{buffer overflow occured on i - th replication} \\ 1, \text{no buffer overflow occured on i - th replication} \end{cases} \tag{3}$$

If the occurrence of overflow is very rare, a large value of N is needed for an accurate estimation of l, the concept of Normalized Variance $\overline{\sigma^2}$ can explain why this is the case:

Let p=$\bar{\ell}_N$

$$\text{var}(p) = \frac{nVar(L(Y_i))}{n^2} = \frac{np(1-p)}{n^2} = \frac{np(1-p)}{n^2} = \frac{p(1-p)}{n} \tag{4}$$

$$\overline{\sigma^2} = \frac{\mathrm{var}(p)}{p^2} = \frac{\frac{p(1-p)}{n}}{p^2} = (\frac{1}{p} - 1)\frac{1}{n} \tag{5}$$

(Note that *L(**Y**ᵢ)* is either 1 or 0, so the Bernoulli distribution is used.) When *p* is a small number, 10^{-6}, even with n= 10^6, the normalized variance is equal to 1. If a variance of 0.1 is desired, n has to be equal to 10^7. This is the reason that CMC simulation requires large number of replications.

Assume *g(**y**)* is another pdf, such that *g(**y**)* dominates *f(**y**)* in the sense of absolute continuity [6], then

$$l = \int L(z)\frac{f(z)}{g(z)}g(z)dz = E_g\{L(Z)\frac{f(Z)}{g(Z)}\} \tag{6}$$

where the *g* in $E_g\{\}$ indicates that the expression is taken with respect to the pdf *g(z)*, , and $\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_N$ are random i.i.d. samples from the pdf *g(z)*.

Based on (4), an alternative estimator to the CMC estimator is the Importance Sampling Estimator:

$$\overline{\ell}^i_N = \frac{1}{N}\sum_{i=1}^{N} L(Z_i)W(Z_i) \tag{7}$$

where *W(z)=f(z)/g(z)* is the likelihood ratio

Now, suppose *g(**y**)=f(**y**,v)*, that is, *g(**y**)* belongs to the same parametric family as the distributions *f(**y**)=f(**y**,v₀)*, then the likelihood ratio $W(\boldsymbol{Z}_i)$ is:

$$W(Z_i) = \frac{f(Z_i, v)}{f(Z_i, v_0)} \tag{8}$$

In another word, we could apply the samples of the pdf *g(z)* into the network, and find out the expected sample performance if we applied the samples of the pdf *f(z)* instead.

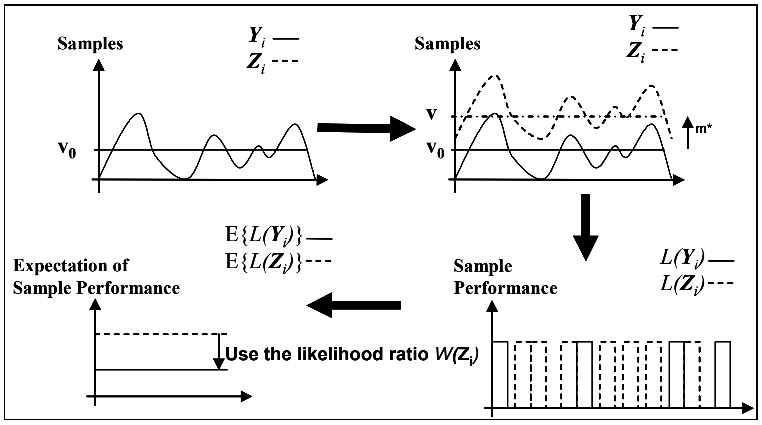


Figure 1. Importance Sampling Summary

As shown in Figure 1, we are interested in $E\{L(\boldsymbol{Y}_i)\}$, yet the network under $\boldsymbol{Y}$ only have a few event occurrences; thus, a large number of replication runs (N) is required. If we can change the parameter of the pdf $f(z,v_o)$ to $f(z,v)$ and make the event occurrences more frequent then we will not need so many replication runs.

In Figure 1, $\boldsymbol{Z}$ is applied into the network instead of $\boldsymbol{Y}$. The sample performance output is recorded from the network. The captured samples of the performance are used to calculate the expectation. With the help of the likelihood ratio, we can find the expectation of the sample performance if $\boldsymbol{Y}$ is applied instead. In conclusion, Importance Sampling shortens the required simulation time.

2.2. Importance Sampling based Traffic Engineering

The basic Importance Sampling technique measures the expected performance of the rare event simulation; it can not estimate the changes in the expected performance with respect to the network parameters(or network inputs). The Importance Sampling based Traffic Engineer (ISTE) [2] approach is able to map the parameters (network inputs) to the performance of the network. For example, ISTE can determine, by reducing the mean rate of an ingress flow, how much will the probability of buffer overflow of the network decrease; thus, determining what the mean rate should be in order to achieve a target probability of buffer overflow.

ISTE is based on the concept of Importance Sampling, however, instead of fixing v_o and change v, ISTE fix v and change v_o. In another word, the current input to the network is modeled by $\boldsymbol{Z}$ () instead of $\boldsymbol{Y}$(). In the simulation, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are of the same parametric family while the only difference between them is their mean rate, by a twisted amount (m^*).

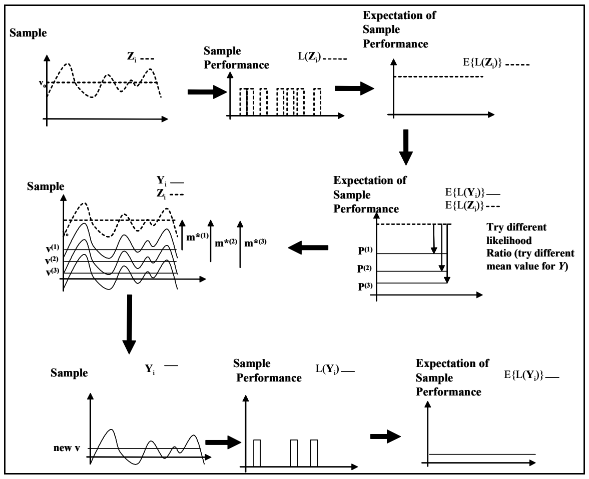


Figure 2. ISTE Summary

Figure 2 is a graphical view of the ISTE approach. We start with the capturing of the sample performance of the current network. By setting the twisted value (m*) to 0, the Importance Estimator becomes a Monte Carlo Estimator and gives the current expectation of sample performance in the network. We then try different values (this process is called "twisting") for the twisted amount ($m*^{(1)}$, $m*^{(2)}$, $m^{*(3)}$,…), thus we will calculate different expectation of performance ($P^{(1)}$,$P^{(2)}$,$P^{(3)}$,…) corresponding to expected network performance under different $\boldsymbol{Y}_i$'s each with different mean values ($v^{(1)}$, $v^{(2)}$, $v^{(3)}$,…). The Importance Sampling theory promises that when those $\boldsymbol{Y}_i$'s are applied to the network, the probabilities of overflow will be the expectations we have calculated. In summary, we have created a mapping between the mean rates of the input process to the expectation of the sample performance of the network.

As shown by equation (5):

$$\overline{\ell}^{i}_{N} = \frac{1}{N}\sum_{i=1}^{N} L(Z_i)W(Z_i)$$

where $L(\boldsymbol{Z}_i)$ is the captured initial sample performance, and N is the number of replication used to capture the initial sample performance, therefore those two terms in the equation are fixed. As mentioned before, $W(z)$ is the likelihood ratio between the pdf $f(z,v)$ and $f(z,v_0)$, and v differ only v_0 by an amount of $m*$ (twisted amount). Twisting is essentially the assigning of different values to $m*$ to find different value for the likelihood ratio which in turn calculates the expected performance, when input processes with different mean values are applied to the network. The twisting operation is therefore a calculation based operation which does not require additional simulation runs, however, as stated in [2]. The performance of this estimator is typically measured by the normalized variance as defined below.

$$\overline{\sigma^2} = \frac{\sum_{i=1}^{N}(L(Z_i)W(Z_i) - \overline{\ell}^{i}_{N})^2 / N /(N-1)}{\overline{\ell}^{i}_{N}} \tag{9}$$

When $m*$ (twisted amount) is small, typically the normalized variance is small [5]. When $m*$ (twisted amount) becomes larger and larger, the normalized variance will become larger in the end. For a network with multiple input sources, if the input process being twisted **is not the major cause of a specific performance metric,** then the prediction of ISTE becomes inaccurate and the normalized variance of the predictor becomes large very fast. This is understandable since normalized variance indicates the accuracy of the estimation, and if the estimation is based on a quantity that has little or no relation to the performance in question, the accuracy of the estimation will not be good. Therefore, once the normalized variance of an estimator for a traffic source or a traffic flow becomes too large, another simulation run will be executed to reduce the variances and the twisting process repeats on another input process (another traffic flow or traffic source) since the previous input process does not have much effect on the network performance anymore[2]. This will result in several additional simulation runs aside from the initial simulation run for the experiment.

3. Fast Importance Sampling based Traffic Engineering

In ISTE [2], for a network with multiple input sources, we twisted one source until the normalized variance was too large. Then we conducted another round of simulation under the twisted traffic load. This allowed us to start twisting another source under a smaller normalized variance. However rerun simulation can still be time consuming because it is under a much lower probability. It limits the potential gain of ISTE in terms of efficiency. To make ISTE more time efficient, it will be wise to improve it so that it requires only the initial simulation run and no extra simulation runs. To do so, we have to be able to twist more than one network input processes without re-running the simulation. The key to achieve this is the concept of dominating source. Network performance is typically dominated by one source under one traffic setting [2]. When we twist a dominating source, it gradually becomes not so dominating and the normalized variance becomes larger at the same time. In the end, the normalized variance becomes very large and the source is not relevant anymore. If we can detect that a dominating source is becoming not so dominating at an earlier stage, we can keep the normalized variance small while we switch to a new dominating source. Then we do not need to rerun simulations every time we switch sources.

Assume $\boldsymbol{Y}$ and $\boldsymbol{X}$ are the inputs to the network, $\boldsymbol{Y}_1,.., \boldsymbol{Y}_N$ are the random i.i.d. samples from the pdf $f(\boldsymbol{y})$, $\boldsymbol{X}_1,.., \boldsymbol{X}_N$ are the random i.i.d. samples from the pdf $h(\boldsymbol{x})$, and $L(\boldsymbol{Y},\boldsymbol{X})$ is the sample performance of the network. The expectation of the sample performance is of our interest:
$l = E\{L(\boldsymbol{Y},\boldsymbol{X})\}$. The expectation is:

$$l = E\{L(Y,X)\} = \int\int L(y,x) f(y) h(x) dy dx \tag{10}$$

Assume $g(\boldsymbol{y})$ and $d(\boldsymbol{x})$ are two another pdf 's, such that $g(\boldsymbol{y})$ dominates $f(\boldsymbol{y})$ in the sense of absolute continuity and $d(\boldsymbol{x})$ dominates $h(\boldsymbol{x})$ in the sense of absolute continuity, then

$$l = \int\int L(z,t) \frac{f(z)}{g(z)} \frac{h(t)}{d(t)} g(z) d(t) dz dt \tag{11}$$

Let $W(z)=f(z)/g(z)$ and $W'(t)=h(t)/d(t)$ be the corresponding likelihood ratios, and $\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_N$ are random i.i.d. samples from the pdf $g(\boldsymbol{z})$, and $\boldsymbol{T}_1,\ldots, \boldsymbol{T}_N$ are random i.i.d. samples from the pdf $d(\boldsymbol{t})$, then

$$l = \int\int L(z,t) W(z) W'(t) g(z) d(t) dz dt \tag{12}$$

Assume the two processes $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent, the new Importance Estimator is:

$$\bar{\ell}_N^{i} = \frac{1}{N} \sum_{i=1}^{N} L(Z_i, T_i) W(Z_i) W'(T_i) \tag{13}$$

As it can be seen, the new equation for the Importance Estimator differs from the original one by only taking into account the new likelihood ratio. Now, suppose $g(\boldsymbol{y})=f(\boldsymbol{y},v)$, $f(\boldsymbol{y})=f(\boldsymbol{y},v_0)$, $d(\boldsymbol{x})=h(\boldsymbol{x},u)$, and $h(\boldsymbol{x})=h(\boldsymbol{x},u_0)$ then the likelihood ratios $W(\boldsymbol{Z}_i)$ and $W'(\boldsymbol{T}_i)$ are:

$$W(Z_i) = \frac{f(Z_i, v)}{f(Z_i, v_0)} \qquad W'(T_i) = \frac{f(T_i, u)}{f(T_i, u_0)} \tag{14}$$

As a result of (11), we can twist one of the several inputs of the network until the normalized variance gets large, which indicates the source is not dominating anymore.. Then we will start twisting the mean rate of another input. This repeats until we reach our target network performance. Therefore, the whole experiment requires only the initial pilot run for the first twisting and no addition simulation runs. Keep in mind that twisting is a calculation based operation, it basically makes predictions on the performance of the network, and it does not require additional simulations.

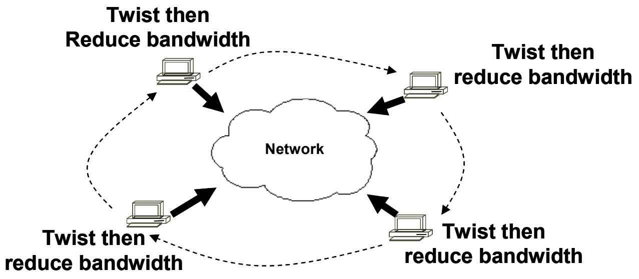

Twist then Reduce bandwidth
Twist then reduce bandwidth
Network
Twist then reduce bandwidth
Twist then reduce bandwidth

Figure 3 FISTE Twisting

In this paper, we propose and compare two twisting approaches as our fast ISTE (FISTE) solution: gradient based twisting and threshold based twisting. In the gradient based twisting approach, a fixed amount of mean rate reduction (a step) is predefined. All the input processes of the network use the same step amount. The input with the smallest gradient of the normalized variance under this step is the dominating source under current traffic setting. Then we twist this source for one step. After the twist, we reevaluate the gradients under the new traffic setting. This repeats until we reach our target performance. In the threshold based twisting, we still test all the inputs and pick the input that is most dominating of the network performance according to the gradients of the normalized variances at the beginning. Once we have picked our input process of the network, instead of reducing it by only a fixed amount (step), we will reduce it until the normalized variance of the twisting process exceeds a threshold. Then we move to finding the new dominating input of the network.

4. Simulation

4.1. Simulation Framework

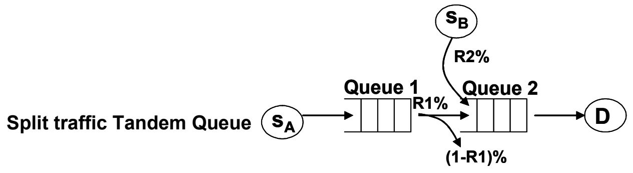


Figure 4. Network Topology

Figure 4 is the network topology used to verify the effectiveness of the FISTE approach. It consists of a tandem queue, two traffic sources with traffic split after the first queue and traffic merge at the second queue. This topology may be simple but all the complex network topologies are some combination of this topology. Traffic Source A models the main ingress network traffic flow while source B models the background traffic flows from other sources in the network

The Lindley equation [7] is used to model the queue behavior.

$$Q_k = (Q_{k-1} + X_k - \mu)^+ \qquad (15)$$

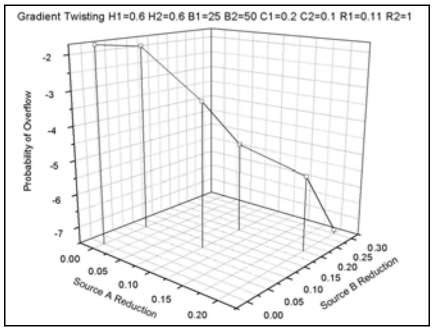
Assume time slot based measurement is used, under the Lindley equation, the queue size depends on the difference between the instantaneous incoming traffic rate (X_k) at time slot k and the queue service rate (μ). At the end of each replication run, we check if the queue size has exceeded the buffer size. If the buffer size has been exceeded, that replication run is considered as a replication run with a buffer overflow event.

Self-similar Fractal Gaussian Noise [8] distribution is used to model the network traffic. Because the self-similar traffic model is used, each traffic source has a Hurst parameter. The Hurst parameter indicates the degree of self-similarity for traffic flows that exhibit long range dependency. The larger the Hurst parameter, the more bursty the traffic is. More bursty traffic source usually generates more faulty events in the network than non-bursty traffic source.

Here are some notations for reading the graphs for our simulation results: B1: Size of buffer in Queue 1 (units/sec). B2: Size of buffer in Queue 2 (units/sec). C1: Service rate in Queue 1 (units/sec). C2: Service rate in Queue 2 (units/sec).H1: Hurst parameter of traffic source A. H2: Hurst parameter of traffic source B. R1: Percentage of traffic entering Queue 2 from Queue 1. R2: Percentage of traffic entering Queue 2 from Source B.

We will be using both gradient twisting FISTE and threshold twisting FISTE to predict the probability of buffer overflow as the mean rate of the traffic sources are reduced. Then, we will reduce the traffic sources by the amount recommended by the two approaches, and use CMC simulation run to verify that the network performance is instead what we have predicted.

4.2. Simulation Results



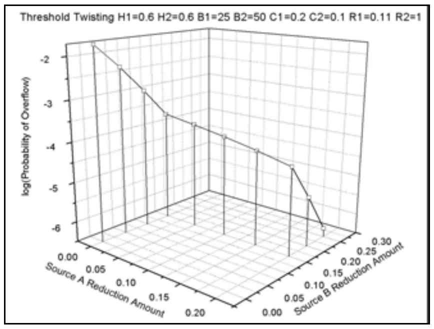


Figure 5. Queue 2 under Gradient Twisting Figure 6. Queue 2 under Threshold Twisting

The FISTE approach can be used on the performance of a single network node, or the performance of an end-to-end path, since *L(**X**,**Y**)* could be the behavior of any part of the network. It should be noted that FISTE, similar to ISTE[2], can tolerate multiple congestion points in the network while most other analytical approaches always assume a single congestion point in the network. This makes FISTE more practical in the real networks than the rest. From figure 5, FISTE shows that if we reduce the mean rate of source A by 0.2 units/second and source B by 0.3, the expected probability of buffer overflow in Queue 2 should be (in logarithmic scale) -7.3834. The CMC simulation showed that if the mean rates of source A and source B are reduced by 0.2 and 0.3, respectively, the measured probability of buffer overflow is -7.3812. As it can be seen, the prediction made by FISTE is accurate. MC simulation was not executed for every point in the figures for comparison. If it was done, the resulting simulation time will be extremely long. The final goal is to reach the target probability of overflow, therefore the intermediate points are less important. Figure 6 also shows the result of the FISTE prediction but using a different twisting scheme. The two twisting schemes arrive at similar conclusion/recommended course of action. The trial simulation run FISTE is 10, 000 replication, and the FISTE normalized variance for the last prediction ($10^{-7.3834}$) is 0.0099. For a CMC simulation to achieve the same kind of accuracy (in terms of the normalized variance), it will require 2.4 billion replication runs:

$$\overline{\sigma^2} = (\frac{1}{p} - 1)\frac{1}{n} \Rightarrow n = (\frac{1}{p} - 1)\frac{1}{\overline{\sigma^2}} = (\frac{1}{10^{-7.3834}} - 1)\frac{1}{\overline{0.00999}} = 2.4 * 10^{9} \tag{16}$$

Compared with the FISTE 10,000 replication with CMC's $2.4*10^9$ replication for there is an astronomical gain in simulation time. This is reflected on the simulation time of the FISTE and MC approach on a Dell Workstation with Intel Xeo 1.7 GHz. The FISTE approach only took 10 minutes while the MC approach took 1 day.

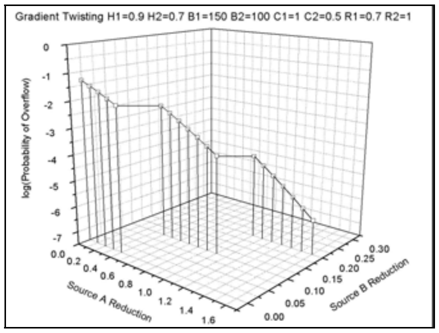


Figure 7. End-to-End Gradient Twisting

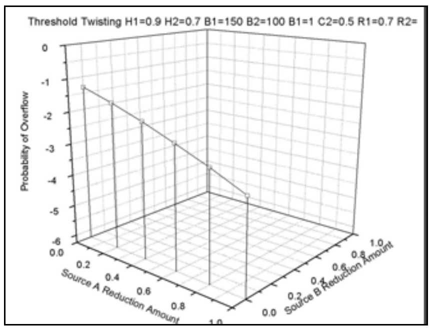


Figure 8. End-to-End Threshold Twisting

Figure 7 and 8 are results for an end-2-end test scenario. Here, a different Hurst parameter is used. A Hurst parameter of 0.9 is used to test if the algorithm still works under bursty traffic. An End-to-end buffer overflow event is defined as an event when there is a buffer overflow in either queue 1 or queue2 or both. Using gradient twisting, as shown in Figure 7, FISTE predicts that when the mean rates of Source A and Source B are reduced by

1.6 units/sec and 0.2 units/sec, respectively, the end-to-end probability of buffer overflow is expected to be (in logarithmic scale) -6.165. Using a CMC simulation run, the measured end-to-end probability of buffer overflow, after the mean rates have been reduced, is -6.392. The FISTE prediction is off by 3.5%, which is tolerable. If threshold based twisting is used instead, FISTE determines that a reduction of 2 units/sec of source A alone should yield a probability of buffer overflow of -6.4929. The CMC simulation run shows that the probability is actually -6.518. Once again, the FISTE prediction is very close to the measured value. Since prediction by gradient-based twisting and threshold-based twisting are both valid, this implies that there different path/course of action to achieve a single target performance, depending on the optimization requirements; one can decide which course of action is the best one.

5. Conclusion

The FISTE approach is a fast, simple, and accurate way of predicting how the performance of a network changes with respect to the network parameters and inputs. The simulation results of this paper show that FISTE can calculate the changes in the probability of buffer overflow of a network as the mean rates of the network ingress traffic decreases. FISTE is even faster than the previous ISTE approach since FISTE requires only a single simulation run whereas ISTE requires several. By using different twisting approaches (step based or threshold based), one can find different ways to achieve the target performance. In the future, given the performance optimization constraints, one can evaluate which course of action is the best.

REFERENCES

1. R. Y. Rubinstein and B. Melamed, Modern Simulation and Modeling, John Wiley & Sons Inc, 1998, pp. 113
2. B. Feng, Importance Sampling Based Internet Traffic Engineering Under Self Similar Traffic Model, Master of Applied Science Thesis, Carleton University, 2005
3. W.E.Leland, M.S. Taqqu, W. Willinger, and D.V.Wilson, On Self-Similar Nature of Ethernet Traffic (Extended Version). ACM/IEEE Transactions on Networking, 2(1):1-15, Feb. 1994
4. V. Paxson and S. Floyd. , Wide Area Traffic: The failure of Poisson Modeling IEEE/ACM Transactions on Networking, 3(3):226-244,1995
5. R. Y. Rubinstein and B. Melamed, Modern Simulation and Modeling, John Wiley & Sons Inc, 1998, pp. 77
6. R. Y. Rubinstein and B. Melamed, Modern Simulation and Modeling, John Wiley & Sons Inc, 1998, pp. 89-107
7. J.W. Cohen. The Single Server Queue, North-Holland, 1982
8. C. Huang, Long range dependent traffic: modeling, simulation, and congestion control, Doctor of Philosophy thesis, Carleton University, Canada, 1997