An End-to-End Performance Inference Technique for Peer-to-Peer Networks

Benjamin Zhong Feng, Changcheng Huang, SCE Dept., Carleton University, Ottawa, Ont. K1S 5B6, Canada
Michael Devetsikiotis, ECE Dept., North Carolina State University, Raleigh, NC 27695-7911 USA

***Abstract*—For voice/video applications that are based on Peer-to-Peer (P2P) models, ensuring the end-to-end Quality of Service (QoS) is crucial, especially if users are paying fees. In this paper we propose an End-to-end Performance Inference Technique (EPIT) that uses a prediction-based approach to map the ingress traffic levels of the P2P network to the end-to-end QoS in the network. Furthermore, by coupling Simulated Annealing (SA) with EPIT, we describe a traffic engineering solution in such a way that the QoS constraints are met while traffic flows into the network are maximized.**

***Index Terms*—Peer-to-Peer, End-to-End QoS, Importance Sampling, Self-Similar traffic model.**

I. Introduction

While many free P2P applications disregard QoS and settle for the current best-effort-approach, other big P2P players do pay attention to the QoS of the service they are offering. For example, Skype is a popular, proprietary software that is based on the P2P model. Its main application is real-time voice chat that makes QoS extremely important. While calls over the Internet are free, Skype does charge a fee for calls to landlines and cell phones. It is well known that during peak hours, the voice quality of Skype conversation degrades, which is likely caused by packet losses or long delays due to high traffic volume.

For the above reason, it is extremely useful to have an approach that can predict the QoS in the network with respect to the ingress traffic levels, when designing or managing a P2P network to deliver a guaranteed QoS. In other words, a mapping between the input traffic and the network performance is desired. While the ingress traffic of the P2P network can be controlled via traffic shaping or call admission control, there are other ways to improve QoS in networks via equipment upgrade, additional bandwidth purchase from providers, changing resource allocation, or modifying routing strategy. There are obvious financial costs attached to equipment and bandwidth purchase. This paper focuses on the ingress-traffic control aspect.

Clearly, less traffic in the network always leads to less congestion and delay, yet can also lead to less revenue: each refused voice call can translate to a loss in fee. Thus, the goal is to achieve minimum traffic reduction while satisfying QoS requirement(s).

Manuscript received March 31, 2008. This work was supported in part by the Carleton University

P2P networks can be classified as *overlay networks*. An overlay network is a network built on top of another network, usually a core provider network. Nodes in the overlay can be treated as being connected by virtual or logical links. Each of which corresponds to an end-to-end path going through many physical links and nodes in the underlying network. Thus, in QoS-enabled P2P networks, the performance of the end-to-end path is crucial.

Over-provisioning the end-to-end paths can deliver guaranteed network performance. However, because many applications (voice/video) can typically tolerate the loss or delay of a small fraction of the traffic, provisioning of statistical QoS guarantees can lead to a more efficient utilization of network resources than worst case based QoS provisioning. Over-provisioning often means more bandwidth purchase that leads to higher cost and less revenue.

It is intuitively obvious that the more traffic is allowed into the network, the higher the utilization will be. Thus, our objective is to use statistical provisioning to maximize ingress traffic while achieving the expected QoS in the P2P network.

Packet loss probability is an important QoS metric. Delay in a network is highly related to packet loss since a small buffer means less delay but higher loss probability. Estimating the probabilities of congestion phenomena, such as packet losses due to buffer overflows and delay, is a very difficult task, especially in view of complicated traffic models (such as the long-range dependent traffic models). This is because traffic flows may lose their original statistical characteristics at the output of a queue. Even though significant progresses have been made both in analytical results and in simulation techniques, as described in Section II, the problem itself is still far from being completely solved.

The End-to-end Performance Inference Technique (EPIT) proposed in this paper essentially predicts the end-to-end overflow probabilities of a path P_{EE}, given the mean rates (bandwidths) of the ingress traffic flows, $v_1,v_2,v_3,..$ In other words, EPIT determines $P_{EE}(v_1,v_2,v_3,..)$ by providing a mapping between the end-to-end QoS and the traffic levels.

In P2P networks, v_i could present the seeding traffic from individual P2P peers. They can also be the aggregated traffic from a number of peers instead of a single peer.

We will demonstrate that our EPIT can be easily modified to work with end-to-end delay related QoS measurement as well as end-to-end buffer overflows.

Returning to the goal of maximizing traffic flow (maximizing utilization) while meeting all the expected end-

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

to-end QoS target (T_{EE}), we combine the Simulated Annealing (SA) approach with EPIT (SA-EPIT) to find the maximum values for $v_1,v_2,v_3,\ldots$ such that $P_{EE}(v_1,v_2,v_3,..)<T_{EE}$.

II. Related Work

The traffic carried by the Internet has shown a vast diversity that poses a significant challenge to all analytical techniques. Extensive measurements of real traffic data [1][2][3], have led to the conclusion that Internet traffic and Ethernet traffic cannot be sufficiently represented by traditional models (e.g., Markovian), which exhibit short range dependency (SRD), but instead possesses long-range dependent (LRD) characteristics that can be more accurately matched by self-similar models.

One of the most notable analytical techniques developed in the early 80's is the Matrix-Geometric solution [4]. However, as the number of nodes in a network increase and the user traffic becomes more complex, this approach quickly becomes intractable. While [5] provides a product-form solution so that each node can be analyzed independently, unfortunately, it is only applicable to specific Markovian traffic models. In the early 90's, large deviation techniques were successfully applied to a single queue under Markovian traffic models [6] [7]. The work in [8] on blocking probability in single network links and the work in [9] on stochastic network calculus also are based on Markovian input and SRD traffic models.

The authors in [10] derived the asymptotic steady-state waiting time for LRD traffic and for single-server queues. In [12], the authors predicted the characteristics of congestion events under a self-similar traffic model. However, the analysis conducted is limited to single queue networks too.

In summary, the existing approaches all have their limitations: dependency on SRD traffic models, or applicable only to single queue networks. Instead, EPIT addresses all those issues.

EPIT is based on the concept of Importance Sampling (IS): Consider a random variable X and let $f(x,v)$ be the true distribution of X, where v is a statistical parameter of the distribution (mean, variance, etc). We wish to estimate P=Pr[$X\in A$]. P can be estimated via direct simulation (also known as the Monte Carlo method): $\hat{P}=\sum_{i=1}^{N} I_A\left(X^{(i)}\right)/N$ where $X^{(i)}$'s are independent samples of X and $I_A(.)$ is an indicator function that equals one if $X^{(i)}$ is in the set A.

Importance Sampling (IS) translates the true probability density $f(x,v)$ to a simulation distribution $f(x,v')$ in order to generate more events and weights these events afterwards with a likelihood ratio: $L(x,v,v')=f(x,v)/f(x,v')$. Intuitively, $f(x,v')$ will make the event $X\in A$ occur more frequently, thus, N does not have to be large for an accurate estimation of P, if P is small. Mathematically:

$$P=\int I_A(x)\frac{f(x,v)}{f(x,v')}f(x,v')dx=E_{f(v')}(I_A(x)L(x,v,v')) \quad (1)$$

The variance of the IS estimator is

$$\sigma^2=\frac{1}{N}\left(\int\left(\frac{I_A(x)f(x,v)}{f(x,v')}\right)^2 f(x,v')dx-P^2\right) \quad (2)$$

Equation (1) can be implemented as

$$\hat{P}=\frac{1}{N}\sum_{i=1}^{N} I_A\left(X^{(i)}\right)L\left(X^{(i)},v,v'\right) \quad (3)$$

where $X^{(i)}$ follows the distribution of $f(x,v')$ instead of $f(x,v)$.

One can vary v', so the variance given in equation (2) is minimized. However, instead of fixing v and varying v', we can fix v' and vary v instead. Thus, we can predict the P as a function of the parameter v using equation (3). This approach is called Importance Sampling Parameter Estimation (ISPE) in this paper, also known as the Response Surface methodology. In summary, based on one simulation run under $v=v'$, $P(v)$ can be predicted for any other values of v. See [13] for more info on IS and ISPE.

ISPE has two critical issues: 1) How to calculate the likelihood ratio for the specific input distribution. Without it, the result will be biased. 2) How to assess the efficiency of the approach: equation (2) clearly shows that as P becomes smaller, the variance becomes larger. The question whether we can find an efficient simulation distribution to compensate for a very small P so the variance does not "blow-up". If ISPE is not efficient, one must utilize a large number of replications (large N) to bring down the variance, which is clearly inefficient. Since EPIT is based on ISPE, those two issues will be addressed in the next section.

III. End-to-end Performance Inference Technique

The End-to-End Performance Inference Technique (EPIT) is based on the classical Importance Sampling Parameter Estimation (ISPE) approach. Therefore, it is important emphasize what separates our approach from the other Importance Sampling based approaches:

1. This paper is the first to apply IS under self-similar traffic model to end-to-end networks with multiple inputs and multiple queues.
2. This paper is the first to apply IS to traffic engineering: maximizing traffic utilization while meeting QoS expectations.

These two points highlight the novelty of our proposed approach. We begin the EPIT discussion with an arbitrary multi-queue multi-source network model. Each host in Figure 1 act both as a traffic source and as a traffic sink, and there are S sources in total. The traffic flow from the traffic sources are modeled as independent random processes $\vec{X}_1,\vec{X}_2..,\vec{X}_S$ with joint density distributions of $f(\vec{x},v'_1)$, $f(\vec{x},v'_2)$, ,…, $f(\vec{x},v'_S)$, where v'_1 $v'_{2..,}$ v'_S are the mean rates (bandwidths) of the traffic flows. Let $\vec{v}'=[v'_1,v'_2,..,v'_S]$. To distinguish random variables and random processes, or rather scalars from vectors, this paper uses the arrow hat notation to separate them.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

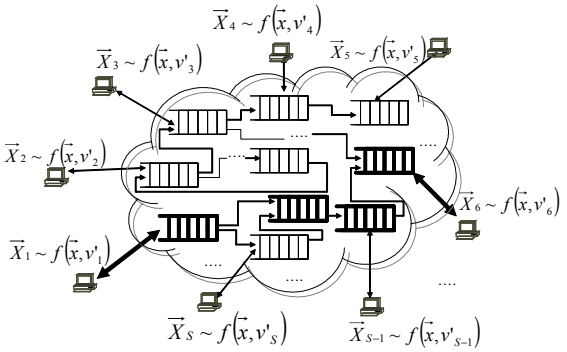

Figure 1 Multi-queue multi-source network

Assume the end-to-end path of interest is between host 1 and host 6, as highlighted. The path spans several queues. Let the set A be a set containing all the realizations (traffic sample traces) from $\vec{X}_1, \vec{X}_2, ..., \vec{X}_S$ such that they will trigger one or more buffer overflows along the end-to-end path if they were injected into the network together. That is, if traffic traces $\vec{X}_1^{(i)}$, $\vec{X}_2^{(i)}$, $..\vec{X}_S^{(i)}$ trigger an end-to-end overflow event, then $I_A\left(\vec{X}_1^{(i)}, \vec{X}_2^{(i)}, ..\vec{X}_S^{(i)}\right) = 1$.

Let $\vec{v} = [v_1, v_2, \ldots, v_S]$ be some other arbitrary traffic level setting. Based on ISPE, one can estimate $P_{EE}(v_1, v_2, .., v_S)$ as:

$$\hat{P}_{EE}(v_1, v_2, .. v_S) = \frac{1}{N}\sum_{i=1}^{N} I_A\left(\vec{X}_1^{(i)}, \vec{X}_2^{(i)}, .., \vec{X}_S^{(i)}\right) \prod_{i=1}^{N} L\left(\vec{X}_i, \vec{v}_i, \vec{v}'_i\right) \quad (4)$$

where a simulation run consists of N independent simulation replications and traffic traces $\vec{X}_1^{(i)}$, $\vec{X}_2^{(i)}, .., \vec{X}_S^{(i)}$ were generated in every replication according to the distributions $f(\vec{x},v'_1)$, $f(\vec{x},v'_2)$, ,…, $f(\vec{x},v'_S)$, respectively.

Equation (4) demonstrates that with a single simulation run under **one specific** traffic level setting $\vec{v}'$ and the buffer overflows on the end-to-end path recorded, equation (4) can infer the end-to-end overflow probability under **any other** arbitrary traffic level settings $\vec{v}$ as long as the likelihood ratio can be calculated. This is why this proposed approach is called End-to-end Performance Inference Technique. This is how EPIT maps the input traffic levels to the QoS of the network.

As stated previously, the likelihood ratio is an important component of EPIT. We have derived $L\left(\vec{X}_i, \vec{v}_i, \vec{v}'_i\right)$ under an FGN traffic model:

$$L\left(\vec{X}^{(i)}, \vec{v}, \vec{v}'\right) = \prod_{k=1}^{S} \frac{f\left(\vec{X}_k^{(i)}, v_k\right)}{f\left(\vec{X}_k^{(i)}, v'_k\right)} = \prod_{k=1}^{S}\prod_{t=1}^{T} L_{j,t} = \prod_{k=1}^{S}\prod_{t=1}^{T} \frac{\exp\left(\vec{X}_k(t)\xi_{k,t}\right)}{\omega_{k,t}} \quad (5)$$

where T is the length of each replication, $\xi_{k,t} = (m_t(v_k) - m_t(v_k'))/\text{var}_{t,k}$, and $\omega_{k,t} = \exp(\xi_{k,t}\ (m_t(v_k) + m_t(v_k'))/2)$.

Quantities $m_t(v_k)$ and $\text{var}_{t,k}$ are the conditional mean and conditional variance for $\overrightarrow{X_k} \sim f(\vec{x},v_k)$ at time t. For more information on how the conditional variance and conditional mean relate to the correlation coefficients and the Hurst Parameter of the FGN process, see [14].

Simulation efficiency is another issue for EPIT. A rarity parameter is a parameter that as it approaches infinity, P approaches zero. Clearly, in a single-queue single source scenario, the buffer size b is a rarity parameter. Let $F = I_A(x)f^2(x,v)/f(x,v')$ in equation (2). One can see that if P^2 decreases faster than F with respect to b, the variance will increase.

Since EPIT is based on ISPE and a closed-form analysis of the multi-queue system under FGN is very difficult, we settled for proving the efficiency of ISPE in a single FGN source-single queue scenario with Theorem 1:

Theorem 1: $R = 2I$ if $v' - v = u\sqrt[2H]{1.5}$ where $\lim_{b\to\infty} \log(F)/b = -R$ and $\lim_{b\to\infty} \log(P)/b = -I$.

The detailed proof can be found in [14] (u is the difference between the service rate and input traffic rate). Intuitively, R is the logarithm rate of decay of F and I is the logarithm rate of decay of P, with respect to the rarity parameter b. Log scale is used, as in [15] where they state that, in the asymptotic settings of large deviations theory, the rare probabilities decays exponentially as the rarity parameter tends to ∞. $R = 2I$ implies that the F is able to "compensate" for the exponentially decreasing P^2 thus preventing the variance from "blowing-up". This proves that ISPE is efficient since N (number of replications) does not have to be increased drastically for an accurate estimation of a decreasing P.

To apply EPIT outside of a simulation environment, instead of generating the traffic traces, we can use a measurement window to capture real traffic traces from the network. However, the capture window has to be large enough to ensure independence among the traces. The indicator function I_A will be replaced by an actual monitor that will report overflow if it were to occur in a router or a network node.

If we redefine the event A as the occurrence of excessive delay in the network, then P_{EE} becomes the probability of excessive end-to-end delay occurring in the network.

IV. Simulated Annealing With EPIT

Assume that under the original traffic level $\vec{v}'$, the end-to-end overflow probability $P_{EE}(\vec{v}')$ on a critical path is above the QoS target T_{EE}: $P_{EE}(\vec{v}') > T_{EE}$. We want to find a suitable reduction setting $\vec{r} = \vec{v}' - \vec{v}$, so that $P_{EE}(\vec{v}' - \vec{r}) = P_{EE}(\vec{r}) \le T_{EE}$. The trivial solution is to let $\vec{r} = \vec{v}'$ which will devoid the network of any traffic. There will be no buffer overflows in the network, but this is clearly not acceptable. We must allow as much traffic in the network as possible to maximize the network utilization and in the same time, satisfy the QoS constraints. Thus, the problem becomes a Non-Linear Programming problem:

The objective function:

$$\min(\Phi(\vec{r})) = \min\left(\sum_{i=1}^{S} C_i r_i + D\left[P_{EE}(\vec{v}' - \vec{r}) - T_{EE}\right]^+\right) \quad (6)$$

where $[x]^+ = 0$ if $x < 0$

The constraint:

$$P_{EE}(v_1, v_2, .., v_S) = P_{EE}(\vec{v}) = P_{EE}(\vec{v}' - \vec{r}) \le T_{EE} \quad (7)$$

The solution space:

$$\vec{r} = [r_1, r_2, \ldots r_S] \text{ where } 0 \le r_1 \le v'_1, \ \cdots, \ 0 \le r_S \le v'_S \quad (8)$$

D is assigned as the cost/penalty for not fulfilling the QoS (overflow probability) requirements in the objective function. It forces the heuristic search process to converge to a solution that can satisfy the constraints.

As stated before, aside from achieving the QoS goals, we also want to maximize the network utilization by maximizing the traffic flow. In other words, minimizing the reduction amounts. Thus, C_i is assigned as the cost of reducing one traffic unit per time unit from the mean rate of customer traffic stream i in the objective function. This allows the user to assign different costs to the different customers' traffic according to criticalness and revenue.

Equations (6)-(8) pose an NP-hard problem and we select Simulated Annealing (SA) [16] as the heuristic search algorithm for this problem. Please refer to [16] for more info on SA. The basic goal of SA is to make a new $\vec{r}$ in each iteration and try to bring each $P_{EE}(\vec{v}' - \vec{r})$ closer to T_{EE} while keeping the cost of reduction low. At the end of the iteration process, the $\vec{r}$ with the lowest cost and $P_{EE}(\vec{v}' - \vec{r}) \le T_{EE}$ becomes the best solution, denoted as $\vec{r}*$.

Instead of using direct simulation or direct measurement in each iteration to find $P_{EE}(\vec{v}' - \vec{r})$, with only one simulation run or one measurement under $\vec{v}'$, EPIT can predict any $P_{EE}(\vec{v}' - \vec{r})$ since $\vec{v} = \vec{v}' - \vec{r}$, as demonstrated in equation (4).

V. Simulation Experiments

This section reports on the simulation experiments used to verify EPIT and SA-EPIT, for P2P networks.

A. Fractional Gaussian Noise Traffic Model

Fractional Gaussian Noise (FGN) process is a self-similar process that exhibits Gaussian characteristics. Since P2P traffic accounts for a huge portion of the Internet traffic (up to 50%) and since the Internet has shown self-similar characteristics, we can also assume aggregated P2P traffic is also self-similar. Furthermore, the central limit theorem guarantees that the aggregation of large number of independent flows will exhibit gaussian property. Thus, we believe using the FGN traffic model is appropriate in this case. Please refer to [14] for more info on the FGN random process.

B. Network Topology

As stated previously, P2P networks are overlay networks. Thus, P2P traffic usually transverses the core provider network that is under the P2P overlay.

The core provider network depicted in Figure 2 is a complicated network based on a real network topology: the National Science Foundation Network (NSFNET). This topology is frequently used in network simulations.

The values for the network parameters listed in Table 1 are generic values: their units are "traffic unit" or "traffic unit/time unit". This is because it is hard to define capacities for a core network as a "typical network". With ever-greater processing power, the data link capacities and router speeds change from year to year. This is why "traffic unit per time unit" could be defined as millions-of-packets-per-second or be defined as giga-bytes-per-minute.

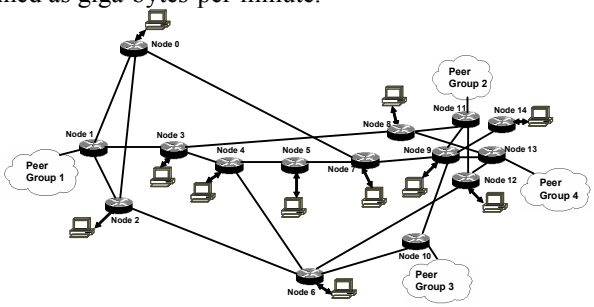

Figure 2 The core provider network topology

Node	Buffer Size (traffic unit)	Service Rate (traffic unit/time unit)	Hurst Parameter of Attached Traffic Source	Mean Rate of Attached Traffic Source(traffic unit/time unit)	Variance of Attached Traffic Source
0	375	9	0.9	1.8	0.1
1	415	10	0.56	5.1	2.2
2	915	16.4	0.74	9.47	4.1
3	252	7.5	0.69	1.37	0.6
4	516	10.5	0.7	4.47	1
5	317	7.6	0.86	3.52	0.16
6	639	16	0.92	5.27	0.2
7	898	17.9	0.59	8	3.8
8	432	6.7	0.83	2.47	0.25
9	612	23.8	0.57	7.97	2.5
10	598	4.6	0.67	2.47	0.5
11	212	11.6	0.72	3.97	0.3
12	391	8.9	0.79	3.8	1
13	576	3.7	0.81	2.67	0.2
14	646	8.3	0.63	5.87	1.8

Table 1 Core provider network parameters (Hurst parameter is an indicator on the burstiness of self-similar/long-range-dependent traffic flows)

The specific values of the network parameters are not important, and EPIT would still work given a different set of settings. To avoid any bias in the network setup, the network parameters (shown in Table 1) are generated randomly according to uniform distributions (Hurst Parameter [0.5 to 0.9], Mean Rate [1 to 10], Variance [0 to 3]) and exponential distributions (Buffers size $\lambda = 500$ and service rate $\lambda = 10$). Due to space limitation, please refer to [14] for other details regarding this network setup.

In the following simulation model, peers are divided into groups according to their geographical locations. Thus, from the core provider network's point of view, each peer in the same peer group will be connected to the same Provider Edge (PE) router. Figure 2 shows the configurations of peering arrangements for torrent 1, embedded in the core network. The other traffic sources represent the traffic not part of the torrent. The arrangements of Torrent 2 and torrent 3 are described in Table 2.

For each torrent setup listed in Table 2, peers in any peer group could be communicating with any peer in the other peer groups. The activity could be seeding/leeching, or it could be voice or video (e.g., Skype). To provide QoS to those applications, the aggregated performance of the end-to-end paths (PE to PE) connecting the peers groups has to meet the QoS target as a whole since we cannot know which end-to-end path the peer user will use.

Assume that the end-to-end overflow probability, P_{EE}, (aggregated) currently exceeds the QoS target T_{EE}. We wish to bring P_{EE} down to T_{EE} by reducing P2P traffic via traffic shaping (P2P traffic can be isolated via port-based analysis of the traffic) or call admission control. We also wish to minimize the reduction in order to maximize the traffic flow.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

We therefore use SA-EPIT.

Torrent	Peer Group	Peer Group PE Node	PE-PE Paths used to connect all peer groups
1	1	Node 1	
	2	Node 11	"1-3-8-13"
	3	Node 10	"13-12-11"
	4	Node 13	"10-9-13"
2	1	Node 3	
	2	Node 10	"3-8-13-9"
	3	Node 11	"9-10"
	4	Node 9	"11-9"
3	1	Node 2	
	2	Node 8	"2-0"
	3	Node 9	"0-1-3-8"
	4	Node 0	"9-7-0"

Table 2 Torrent configurations PE-to-PE Paths in the torrents

Each iteration in Figure 3 corresponds to one potential solution $\vec{r}$. EPIT predicts the corresponding P_{EE} for that particular $\vec{r}$ so SA can calculate the total cost for that $\vec{r}$. Normally, this is where SA moves onto the next iteration. However, since this is the first time EPIT is proposed, we have to demonstrate that the EPIT prediction is accurate. A direct simulation is performed for that $\vec{r}$ to measure its P_{EE}. Those measurements are graphed in the solid data line of Figure 3. Figure 3 clearly shows that EPIT predictions match the measurements. This validates our EPIT approach. The process is repeated on torrent 2 and 3 with similar results. Due to space limitation, those figures are omitted here.

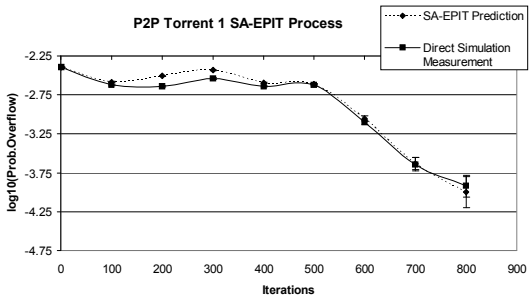

Figure 3 P_{EE} trajectory for torrent 1 as SA-EPIT iterates through different soultions to bring P_{EE} to T_{EE}

$\vec{r}^*$	Torrent 1	Torrent 2	Torrent 3	$\vec{r}^*$	Torrent 1	Torrent 2	Torrent 3
r_0	0.01	0.01	0.02	r_8	0.15	0.17	0.1
r_1	0.12	0.14	0.09	r_9	0.09	0.15	0.2
r_2	0	0.13	0.02	r_{10}	0.02	0.13	0.03
r_3	0.14	0.16	0.13	r_{11}	0.14	0.1	0.12
r_4	0.07	0.04	0.05	r_{12}	0.01	0.03	0.12
r_5	0.05	0.08	0.14	r_{13}	0.05	0.1	0.12
r_6	0.01	0.04	0.04	r_{14}	0.01	0	0.1
r_7	0	0.19	0.04				

Table 3 SA-EPIT solution for P2P experiments, comprised of the reduction amounts for the mean rates of the 15 traffic sources

At the end of the iteration process, SA-EPIT finds the solution $\vec{r}^*$ that has the lowest reduction cost among all other solutions that meet the QoS requirement. We do not have space to list the cost of the other solutions, but the SA objective function guarantees that $\vec{r}^*$ is the solution to realize maximum traffic flow.

As shown in Table 4, the new overflow probabilities on those end-to-end paths used by P2P torrents all meet their respective targets. Therefore, the effectiveness of SA-EPIT is verified.

Torrent	Original P_{EE} (log scale)	T_{EE} (log scale)	New P_{EE} (log scale) after reductions
1	-2.2612	-4	-4.08497
2	-2.5003	-4.25	-4.267819
3	-1.9512	-3.5	-3.5267

Table 4 New overflow probabilities after reductions are made

VI. CONCLUSION

We have proposed the End-to-end Performance Inference Technique in this paper that can infer end-to-end performance of a network with respect to different traffic levels. This is useful for QoS sensitive P2P applications (voice/video) where the service quality will degrade if there is too much traffic in the network. Furthermore, the Simulated Annealing coupled EPIT approach finds a solution that can meet the QoS requirement while maximizing the network utilization.

In our future work, we shall perform comparisons between the proposed approach and other traffic prediction/control for P2P networks, such as structured/coordinated peer-to-peer systems. We shall also apply the proposed method in different P2P scenarios to test its effectiveness and efficiency.

REFERENCES

[1] LELAND, W., TAQQU, M., WILLINGER, W., and WILSON D.V. 1994. On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Trans. on Networking*, 2(1), 1-15.

[2] PAXSON, V. and FLOYD, S., 1995. Wide-area traffic: the failure of Poisson modeling, IEEE/ACM Transactions on Networking, 3(3), 226-244.

[3] MANDELBROT. B. B.,1983. The Fractal Geometry of Nature W. H. Freeman, New York, NY.

[4] NEUTS. M. F. 1981. "Matrix-Geometric Solutions in Stochastic Modeling: An Algorithmic Approach," *Johns Hopkins University Press*.

[5] WALRAND. J. 1988. *An Introduction to Queueing Networks* Prentice Hall, New York, NY.

[6] KESIDIS. G., WALRAND. J., and CHANG. C.S.,1993. Effective Bandwidth for Multiclass Markov Fluids and Other ATM Sources. *IEEE/ACM Trans. Networking* 1(4). 424-428

[7] De VECIANA. G., and KESIDIS. G., 1996. Bandwidth Allocation for Multiple Qualities of Service Using Generalized Process Sharing, *Transaction on Information Theory*, 42(1), 268-272

[8] MICHAEL, M., R¨UDIGER, M,. JOACHIM C. 2006. Capacity Over-provisioning for Networks with Resilience, *In Proceedings of SIGCOMM'06*, Pisa, Italy, 87-98.

[9] Y., JIANG, 2006. A Basic Stochastic Network Calculus. *In Proceedings of SIGCOMM'06*, Pisa, Italy. 123-134.

[10] DUFFIELD. N. G. and O'CONNELL. N., 1993. Large Deviations and Overflow Probabilities for the General Single-Server Queue, with Applications. *Technical Report DIAS-STP-93-30, Dublin Institute for Advanced Studies*.

[12] JIN, Y.,; BALI, S.; DUNCAN, T.E.; FROST, V.S. 2007 Predicting Properties of congestion Events for a Queueing System With fBm Traffic, *IEEE/ACM Transactions on Networking*. 15(5). 1098 – 1108

[13] MADRAS, N,. 2002. *Lectures on Monte Carlo Methods*, American Mathematical Society, Providence, Rhode Island.

[14] FENG, B., End-to-End Overflow Probability Inference Technique (EOPIT)An Overlay Network QoS Approach under Self-Similar Traffic Model, *Doctor of Philosophy thesis Proposal*, Carleton University, Canada,2008http://www.sce.carleton.ca/~zmbfeng/PHD_thesis_proposal_zmbfeng.pdf

[15] SADOWSKY, J.S. BUCKLEW, J.A.,1990, On large deviations theory and asymptotically efficient Monte Carlo Estimation, *IEEE Transactions on Information Theory*, 36(3), 579-588

[16] KREHER, D. L., and STINSON, D. R., Combinatorial Algorithms. CRC Press. New York, NY, 1999.