On Reducing Response Time for VPN Traffic

Idris A. Rai, Guillaume Urvoy-Keller, Ernst W. Biersack
Technical Report TR-02.03.1
Institut Eurecom
2229, route des Crêtes
06904 Sophia-Antipolis, France
{rai,urvoy,erbi}@eurecom.fr

***Abstract*— A successful migration from traditional private networks (PNs) to virtual private networks (VPNs) depends on the ability of VPNs to offer similar quality of service (QoS) guarantees. To approach this goal, we propose to use priority scheduling based on flow size differentiation at the edge devices of the VPN. We evaluate the shortest elapsed time (SET) scheduling policy and compare it to different scheduling policies for both, high and low variance flow size distributions. The results show that SET is a more suitable policy than the traditional FIFO policy in terms of minimizing the response time. In particular, SET offers significantly lower mean response time for flow sizes with a high variance than flow sizes with a low variance. We also propose an implementation of SET for VPN customer edge (CE) devices.**

I. Introduction

A Virtual private network (VPN) is a cost effective method to emulate the characteristics of a private network (PN) over the Internet. VPNs are achieved by constructing a communication environment based on a controlled segmentation of a shared network infrastructure. IPSec offers a reliable privacy level to VPNs and encapsulation enables a virtual topology to be built on top of the existing shared network infrastructure. Therefore, in addition to being cheaper than PNs, VPNs allow dynamic topology, and can be easier to plan and configure than PNs. To completely emulate a private network however, a VPN should provide performance assurance, i.e., Quality of Service (QoS), similar to PNs.

The topology of a VPN can be static or dynamic. Static topology VPNs sustain a fixed number of endpoints. On the other hand, dynamic topology VPNs support mobile users, therefore the number of endpoints of these VPNs can vary. In both cases however, the VPN customer or provider knows the existing VPN endpoints, which makes it easier for QoS VPNs provisioning and management. In [7], [16], a hose model that enables VPN customers to specify service level specifications (QoS requirements) per VPN endpoints is presented. In [7], the model is used for management of VPN resources and in [16], algorithms to provision a VPN in the hose model are proposed.

There are several studies that investigate QoS provisioning for VPNs using the same frameworks proposed so far for QoS provisioning in the public network, i.e., Diffserv, MPLS, Bandwidth Brokers, Traffic Engineering, and QoS routing [13], [17], [21], [9], [3]. However, most of these frameworks have not been implemented even in the public Internet. In general, provisioning and management of QoS in VPNs is an area that is undergoing rapid evolution.

We envisage that the difficulties posed by provisioning and managing QoS for VPNs will make the cost of the service high. Therefore, VPN customers will seek to optimize their network resources by appropriately managing their customer edge (CE) devices. We focus on this issue at the customer edge (CE) devices of a VPN assuming that the customer service level specifications (SLSs) are guaranteed in the core of the network by a VPN service provider [1], [19], [17].

The work of this paper is motivated by the evidence that the Internet traffic is highly variant. That is, it consists of many small flows and more than half of the traffic load is constituted by less than 1% of all traffic flows. This has been referred to as the heavy-tailed property [11], [2], [5]. Downey shows in [6] that the Internet dataset complies better to a lognormal distribution than a heavy-tailed distribution. In particular, the results of the recent work in [8] reveal that it is not easy to decide whether the Internet traffic can fit heavy-tailed distributions or not. The paper asserts that distribution used in [2] to model the Internet traffic flows, which we also use in this paper, does not have heavy-tailed property, rather it has a high variance property. This result shows that the Internet traffic model is not necessarily heavy-tailed rather it fits many distributions that have high *coefficient of variability* (CoV). The CoV, defined as the ratio of the standard deviation to the mean of a distribution, is a common metric to measure the variability of a distribution.

The evidence of a high CoV for Internet traffic can be used to minimize response times of traffic flows. Shortest remaining processing time (SRPT) is known as an optimal scheduling policy since it minimizes the mean response time by giving preference to small flows [20]. In [2], SRPT is compared to processor sharing (PS) and it is proven that SRPT does not significantly penalize very large flows when the traffic flows have a high coefficient of variability. The implementation of SRPT, however, is limited to traffic flows whose sizes are known in advance like Web files in a Web server [10].

The sizes of VPN flows are not known a priori. Therefore, we analyze another scheduling policy known as shortest elapsed time (SET). In SET, the processor gives service to a flow in the system that has received the least service so far. Therefore, an implementation of SET only needs to know the elapsed service time of all flows rather than their sizes. In this paper, we show that the penalty (in terms of increase in response time) experienced by the largest traffic flows depends on the underlying job size distribution. In particular, we demonstrate that for some high variance distributions only less than 1% of the largest jobs experience higher response times as compared to FIFO or PS.

We compare SET to PS, FIFO, and SRPT. The comparison of SET to FIFO is required because it is FIFO that is currently implemented in Internet routers. We compare SET with PS to analyze its fairness, whereas its comparison to SRPT shows how close to the optimal policy SET does perform.

The paper is organized as follows: we define the scheduling policies and traffic models and present the assumptions on the VPN service in the core of the network in Section II. We investigate the benefits of SET over FIFO in Section III. In Section IV, we present numerical results that compare SET, SRPT, PS, and FIFO for high and low variance traffic workloads. We propose an implementation architecture for SET in Section V and conclude the paper in Section VI.

II. VPN Model

In this section, we define our VPN model in terms of the scheduling policies that we consider and present the assumptions for the traffic model and the VPN service in the core of the network.

A. Scheduling Policies

We propose the SET policy at flow level of the traffic, where a flow is defined as a set of packets with the same 5-tuples, `source address, destination address, source port, destination port, and protocol type,` that arrive close in time. To analyze queuing theory of SET however, it is easier to consider that all packets of a flow arrive at the same time instant at the server. We term this bulk arrival as a job.

All active jobs receive an equal share of service in PS scheduling policy. The objective of PS is to assure fairness among active jobs. SET and SRPT are priority based scheduling policies that favor small job sizes. SRPT is well-known optimal policy that minimizes mean response time [20]. In SRPT, an arriving job to the system receives service immediately if it requires the least service among all the jobs in the system; otherwise it waits if there is at least one job in the system that requires less service. SRPT requires that the job sizes are known a priori while SET does not require the knowledge of job sizes. In SET, an arriving job receives full service immediately until its elapsed service time equals the amount of service given to the job that has received the least service. At this point the 2 jobs will share the processor equally until their elapsed service equals the amount of service given to the job that has received the second least service and so on. No job receives service if there is another job in the system with a smaller elapsed service.

Let the average job arrival rate be λ. Assume a job size distribution X with a probability mass function $f(x)$. The abbreviation c.f.m.f.v is used to denote continuous, finite mean and finite variance. Given the cumulative distribution function as $F(x) \triangleq \int_0^x f(t)dt$, we denote the survivor function of X as $F^c(x) \triangleq 1 - F(x)$. We define $m_2(x)$ as $m_2(x) \triangleq \int_0^x t^2 f(t)dt$ and $m_2 \triangleq m_2(\infty)$ is the second moment of the job size distribution. The load of jobs with sizes less than or equal to x is given as $\rho(x) \triangleq \lambda \int_0^x t f(t)dt$. The **mean conditional response time** is defined as $E[T(x)] \triangleq E[(T|X = x)]$ and the **mean response time** is defined as $E[T] \triangleq \int_0^\infty E[T(x)]f(x)dx$. The expressions of $E[T(x)]$ for M/G/1/SET, M/G/1/FIFO, M/G/1/SRPT, and M/G/1/PS are given in [14] as:

$$
\begin{aligned}
E[T(x)]_{SET} &= \frac{\lambda(m_2(x) + x^2(1-F(x)))}{2(1-\rho(x)-\lambda x(1-F(x)))^2} + \\
&\quad \frac{x}{1-\rho(x)-\lambda x(1-F(x))} \\
E[T(x)]_{FIFO} &= \frac{\lambda m_2}{2(1-\rho)} + x \\
E[T(x)]_{SRPT} &= \frac{\lambda(m_2(x) + x^2(1-F(x)))}{2(1-\rho(x))^2} + \\
&\quad \int_0^x \frac{1}{1-\rho(t)} d(t) \\
E[T(x)]_{PS} &= \frac{x}{1-\rho}
\end{aligned}
$$

The **mean conditional slowdown** of a job of size x is defined as $E[S(x)] \triangleq \frac{E[T(x)]}{x}$. Given the definition of $E[S(x)]$ for any two scheduling policies A and B we have the $\frac{E[T(x)]_A}{E[T(x)]_B} = \frac{E[T(x)]_A/x}{E[T(x)]_B/x} = \frac{E[S(x)]_A}{E[S(x)]_B}$.

B. Traffic Models

Traffic measurements suggest that the Internet traffic exhibits a high variability at different traffic levels, i.e., sessions, connections, and flows. That is, more than half of the traffic load is comprised by less than 1% of the largest jobs. This attribute has been observed for sizes of files in web sites, Unix systems, ftp transfers, the size of transmission duration, I/O times, and Unix CPU process requirements [11]. Hence, we emphasize the results for the VPN traffic model with a high CoV value.

In this paper, we compare the performance of SET to FIFO, PS, and SRPT for high variance and low variance traffic models. We use the bounded Pareto distribution $BP(a, P, \alpha)$ (where a and P are the minimum and maximum job sizes and α is the exponent of power law) as a typical example of high variance empirical workloads and the exponential distribution for low variance empirical workloads. The density functions of the bounded Pareto $f(x)_{BP}$ and the exponential distributions $f(x)_{Exp}$ are given as:

$$
\begin{aligned}
f(x)_{BP} &= \frac{\alpha a^\alpha}{1-(a/p)^\alpha} x^{-\alpha-1}, \ a \le x \le p, \ 0 \le \alpha \le 2 \\
f(x)_{Exp} &= \mu e^{-\mu x}, \qquad x \ge 0
\end{aligned}
$$

The BP distribution can have a very high CoV, whereas the exponential distribution has a CoV value of 1. Throughout this paper, we consider the $BP(332, 10^{10}, 1.1)$ with a mean value of 3000 and the exponential distribution with a mean of $2.3 * 10^3$. The CoV of the BP considered is 283.9 and it is always 1 for the exponential distribution. We visualize the variability of these distributions by plotting their *mass-weighted distributions* $F_w(x)$ as a function of their cumulative distribution functions $F(x)$ in Figure 1. The mass weighted function is defined in [5] as $F_w(x) \triangleq \frac{\int_{-\infty}^x u dF(u)}{\int_{-\infty}^\infty v dF(v)}$. If plotted against $F(x)$, $F_w(x)$ yields

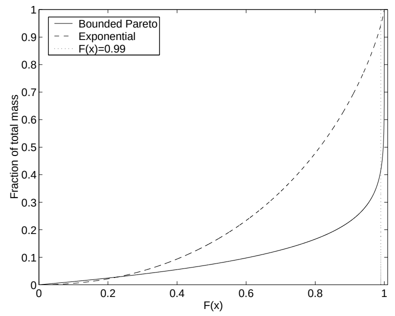


Fig. 1. *Fraction of the total mass for $BP(332, 10^{10}, 1.1)$ and exponential distribution as a function of their cumulative distribution functions at load* $\rho = 0.9$

a fraction of the total mass constituted by all jobs whose size are less than or equal to x. Observe in Figure 1 that the fraction of the total mass comprised by 1% of the largest jobs is more than 50% for the $BP(332, 10^{10}, 1.1)$ distribution. For the exponential distribution, 1% of the largest jobs comprise less than 10% of the total mass.

C. VPN Core Service

It is proposed in the literature [1], [17], [19], [21] that the QoS requirements in the core network can be guaranteed by a VPN service provider (for instance using the MPLS technology). In this paper, we propose deploying the SET scheduling policy in the CE devices and assume a VPN service with guaranteed bandwidth in the core. Hence, any reduction in delay due to the SET scheduling policy at the edges of the VPN will directly result in a reduction of the end-to-end delay seen by the VPN traffic. In the next section, we present general analytical results to compare SET with FIFO.

III. Comparison of SET to FIFO

FIFO is the scheduling policy currently implemented in the routers. To evaluate the benefits that SET can offer to the VPN users, we must compare SET with FIFO for high and low variance distributions. In ([4], pp. 188), the relation between the mean waiting times of FIFO and PS is given by the expression

$$E[W]_{PS} = E[W]_{FIFO} - \frac{\rho[C^2(X) - 1]}{2(1-\rho)} E(X) \qquad (1)$$

where C(X) is the coefficient of variability (CoV) and $E(X)$ is the mean job size. The mean response time, $E[T]$, is given as $E[T] = E[W] + E(X)$. Using this relation in Equation (1) we obtain the same relation between the mean response times of the PS and SET policies:

$$E[T]_{PS} = E[T]_{FIFO} - \frac{\rho[C^2(X) - 1]}{2(1-\rho)} E(X). \qquad (2)$$

In ([18], Corollary 1), we proved the following relation between the mean response time of SET and PS:

$$E[T]_{SET} \leq \frac{2-\rho}{2(1-\rho)} E[T]_{PS} \qquad (3)$$

$$E[T]_{SET} \leq \frac{(2-\rho)}{2(1-\rho)} E[T]_{FIFO} - \frac{\rho(2-\rho)[C^2(X) - 1]}{4(1-\rho)^2} E(X) \qquad (4)$$

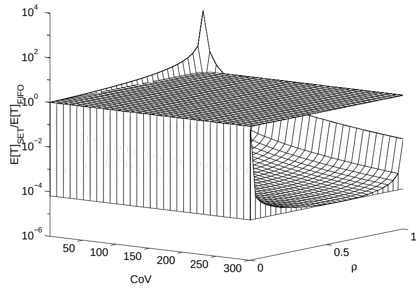


Fig. 2. *Upper bound on the mean response time ratio $\frac{E[T]_{SET}}{E[T]_{FIFO}}$, as a function of load ρ and coefficient of variability*

Replacing $E[T]_{PS}$ in Equation (3) by the right hand side of Equation (2) allows us to bound the mean response time of SET as follows:

We note that the bound of the mean response time of SET is a function of the mean response time of the FIFO policy, the load ρ, and the CoV. This bound is interesting since it enables us to compare the performance of SET relative to that of FIFO for a large range of distributions. Figure 2 shows the upper bound on the ratio of the mean response time of SET to the mean response time of FIFO as a function of $CoV \geq 1$ and load $\rho < 1$. The horizontal plane at in Figure 2 allows to identify the CoV and load values for which SET offers a lower or a higher response time than FIFO. We observe that SET has a higher response time than FIFO for distributions with a CoV close to 1. For CoV close to 1 the mean response time (resp. slowdown) of SET increases with increasing load. On the other hand, the mean response time of SET is lower than that of FIFO for high variance distributions at all load values. For a given load ρ the ratio $\frac{E[T]_{SET}}{E[T]_{FIFO}}$ decreases with increasing CoV.

We now compare SET to FIFO in terms of the mean conditional slowdowns as a function of job sizes for the specific BP and exponential distributions that we consider. Recall that in contrast to SET, FIFO favors large jobs since a job in service the service under the FIFO policy is not interrupted until it leaves the system. As a result, small jobs experience very high slowdowns under FIFO as compared to their slowdowns under SET for both distributions considered. Observe these facts from the slowdown ratio between FIFO and SET in Figure 3. The ratio is above 10^4 for about 99% of the job sizes in case of the BP. Generally, the ratio is lower for the exponential distribution. However, for exponentially distributed job sizes about 80% of the jobs have lower slowdowns under SET than under FIFO. Hence, for traffic with a high CoV, SET greatly improves the slowdown (resp. response time) of many jobs as compared to FIFO. Note in Figure 3 that less than 1% of the largest jobs suffer a higher slowdown under SET as compared to FIFO for the BP distribution (and about 20% for the exponential distribution). These results confirm that SET significantly improves the response time of the VPN traffic.

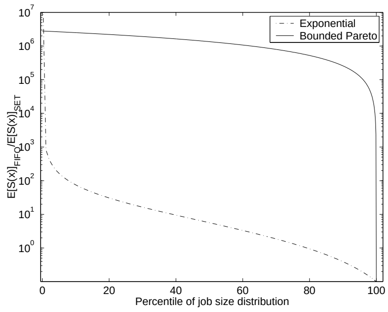


Fig. 3. $\frac{E[S(x)]_{FIFO}}{E[S(x)]_{SET}}$ *for* $BP(332, 10^10, 1.1)$ *and* $Exp(2.3 * 10^3)$ *as a function of percentiles of job sizes, at load* $\rho = 0.9$

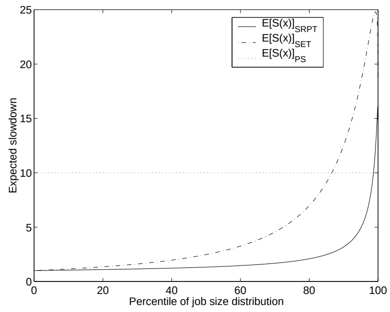
IV. Comparison of SET to PS and SRPT

In this section, we present numerical results that compare the SET policy to the SRPT and the PS policies assuming M/G/1 queuing model. The objective is to elaborate on the following questions for the considered distributions: 1) How close is SET to the optimal policy SRPT in terms of response time and slowdown and 2) How much better is SET than PS in terms of slowdown and response time for a given job size and what percentage of jobs are penalized under SET and the degree of penalty they experience.

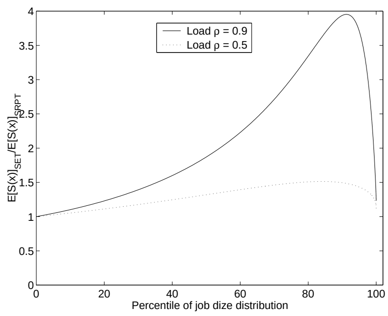
SET ensures an intermediate performance between SRPT and PS, and for a given job size it achieves conditional response time and slowdown much similar to SRPT regardless of the job size distribution, see Figures 4(a) and 5(a). The conditional mean response times and slowdowns for SET and SRPT are closer to each other for the BP distribution job sizes than for the exponential job sizes. Observe in Figure 5(b) that the slowdown ratio between SET and SRPT stays almost constant and close to 1 as opposed to Figure 4(b), where the ratio is close to 4 at load $\rho = 0.9$.

SET favors more small jobs for the BP than for the exponential job sizes. This is to be expected because the BP distribution, as an example of a high variance traffic distribution, has many small job sizes and a few very large job sizes, which establishes the right pattern for scheduling policies such as SET that give preference to small job sizes. Note in Figure 5(a) that 99% of jobs have lower response times under SET than under PS, at load $\rho = 0.9$. Figure 4(a) depicts the slowdowns under the SET, SRPT, and PS policies for exponential job sizes. More than 10% of the largest jobs are penalized and have a higher slowdown under SET than under PS. It is worth noting from the figures that for the exponential job sizes, about 2% of the largest jobs have higher slowdowns under SRPT than under PS, while for the BP distribution all jobs have a smaller slowdown under SRPT than under PS. In [2] the comparison of SRPT with PS yields the same results for this specific BP distribution. This shows that there exist traffic distributions for which all workloads can do better under SRPT than under PS.

In general, we notice that SET effectively favors small jobs at the expense of a slight penalty for the very few largest jobs. The penalty encountered by the large jobs under SET varies with the job size distribution. Note in Figure 4(a) that the maximum slowdown value of the exponentially distributed job sizes under SET is about 2.5 times larger than the slowdown under PS,



(a) $E[S(x)]$ at load $\rho = 0.9$



(b) $\frac{E[S(x)]_{SET}}{E[S(x)]_{SRPT}}$

Fig. 4. *Performance of the policies for* $Exp(2.3 * 10^3)$ *as a function of percentiles of job size distribution*

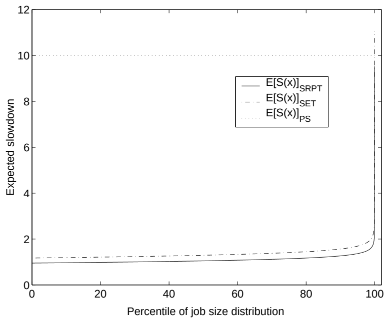
whereas Figure 5(a) demonstrates that for the BP job size distribution, the maximum slowdown under SET is only about 1.1 times larger than the slowdown under PS. Also, from the above results the response time and the slowdown offered by SET are lower for the job sizes with a high CoV than for job sizes with a low CoV. In the next section, we present an implementation architecture for SET scheduling policy for CE devices.

V. SET implementation architecture

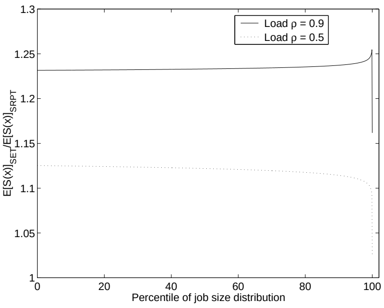
In this section, we discuss an implementation architecture for SET at flow level. SET gives preference to short flows by giving service to the flow that so far has received the least service of all. Hence, its implementation at flow level requires at all time instants the knowledge of the elapsed service of each flow. In [15], SET with finite size quanta is discussed. Here, we demonstrate that it is possible to implement a finite size quanta SET without an infinite number of queues. Indeed, we only need a number of queues equal to the number of active flows, which should be moderate at the CE device.

Each flow is assigned a timer, which is set to a value T_{th} upon the arrival of each packet of the flow. The role of the timer is to determine if a flow is no more active. If the duration between two subsequent packets with the same 5-tuples is greater than T_{th}, the timer is said to have expired and the arriving packets with the same 5-tuples are considered as a new flow.

The implementation architecture of SET that we propose consists of a per-flow classifier, per-flow queuing discipline and



(a) $E[S(x)]$ at load $\rho = 0.9$



(b) $\frac{E[S(x)]_{SET}}{E[S(x)]_{SRPT}}$

Fig. 5. *Performance of the policies for $BP(332, 10^{10}, 1.1)$ as a function of percentiles of job size distribution*

a database management queue (DMQ) as seen in Figure 6. The classifier creates a queue for each newly arriving flow. It also counts and assigns all subsequent packets of the flow in its queue. In addition, it is responsible to delete the queue if its timer expires. Per-flow packet count classifiers were considered in the literature for IP switching and MPOA, see [12] for more details.

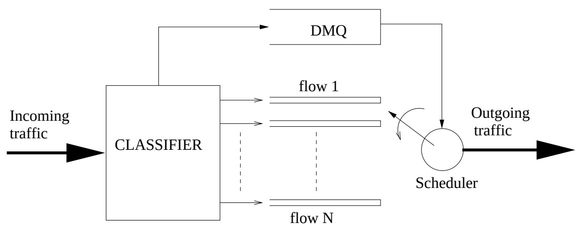


Fig. 6. *An implementation of SET*

The DMQ stores one dmq-identifier per flow. A dmq-identifier is a data structure to identify the state of each flow. The elements of the dmq-identifier include a pointer to memory address of each flow, the number of packets in the queue, a packet counter, and a binary flag b. The pointer to the memory address is created by the classifier. Moreover, the classifier increases the number of packets field in the dmq-identifier every time a new packet of a flow arrives. Packet counter represents the number of packets of a flow that have been served by the scheduler. This can easily be obtained from the scheduler. Therefore, the scheduler is responsible for updating the packet counter and the number of packet fields in the dmq-identifier at the end of each service to a flow. Dmq-identifiers are sorted in an increasing order of the packet counter values. Therefore, a queue receives service if its dmq-identifier is at the head of the DMQ and its corresponding number of packets field is nonzero. The dmq-identifier of the flow that received service is placed at its appropriate position in the DMQ after its packet counter has been updated.

The scheduler performs the following operations: it scans the DMQ for the first entry for which there are packets to serve, services the corresponding queue for a duration equivalent to a quantum of service and updates the information in the DMQ.

It is possible that the timer expires while there are still packets in its corresponding queue of a flow waiting for service. The deletion of this queue must be postponed until the queue is empty. Therefore, when the timer of a flow expires, the classifier first checks the number of packets field in the DMQ before deleting the queue and erasing its dmq-identifier in the DMQ. If the number of packets field of a flow is nonzero, the classifier sets the binary flag b to 1. This indicates to the scheduler that it must delete the queue and erase its dmq-identifies in the DMQ having serviced all packets from it. Otherwise the flag value is always 0 and the classifier is responsible to delete the corresponding queue.

In conclusion, we see that SET can be implemented for VPN edge devices, where the number of flows is reasonably low.

VI. Conclusion

We propose SET as a scheduling policy to reduce the response times experienced by VPN customers. SET's scheduling mechanism introduces flow size differentiation at the CE devices of an enterprise network. While SRPT has been proposed with this objective, its implementation is limited to environments where flow sizes are known in advance. In this paper, we analyze and evaluate the SET scheduling policy and compare it to the SRPT, the PS, and the FIFO scheduling policies. We demonstrate the performance benefits of SET over the PS and FIFO scheduling policies when the traffic distribution exhibits a low CoV (coefficient of variability) or a high CoV. Our analytical and numerical results reveal that the performance of SET in terms of slowdown and response time of traffic flows is better for flow distributions with high CoV than for flow distributions with low CoV. For the distribution with a high CoV that we considered, SET reduces the response times and slowdowns of more than 99% flows at the expense of a small increase in response time and slowdown for less than 1% of the largest flows. Finally, we propose an implementation architecture of SET scheduling policy for VPN CE devices, which demonstrates that an implementation of SET is feasible at flow level.

References

[1] H. F. Badran, "Service Provider Networking Infrastructures with MPLS", In *Proceedings of the Sixth Symposium on Computers and Communications,* pp. 312–318, Hammamet, Tunisia, 2001.

[2] N. Bansal and M. Harchol-Balter, "Analysis of SRPT Scheduling: Investigating Unfairness", In *Sigmetrics 2001 / Performance 2001,* pp. 279–290, june 2001.

[3] T. Braun, M. Gunter, and I. Khalil, "Management of Quality of Service Enabled VPNs", In *IEEE Communications Magazine*, pp. 3–12, May 2001.

[4] E. G. Coffman and P. J. Denning, *Operating Systems Theory*, Prentice-Hall Inc., Englewood Cliffs, 1980.

[5] M. E. Crovella, "Performance Evaluation with Heavy Tailed Distributions", In *Job Scheduling Strategies for Parallel Processing 2001 (JSSPP)*, pp. 1–10, 2001.

[6] A. B. Downey, "The structural cause of file size distributions", In *Proc. MASCOTS 20001*, August 2001.

[7] N. G. Duffield et al., "A Flexible Model for Resource Management in Virtual Private Networks", In *Sigcomm'99*, pp. 95–108, 1999.

[8] W. Gong et al., "On the Tails of Web File Size Distributions", In *Proc. of 39-th Allerton Conference on Communication, Control, and Computing*, October 2001.

[9] M. Gunter, T. Braun, and I. Khalil, "An Architecture for Managing QoS-enabled VPNs over the Internet", In *IEEE Local Computer Networks*, pp. 122–131, 1999.

[10] M. Harchol-Balter et al., "Implementation of SRPT Scheduling in Web Servers", In *IPDS 2001*, pp. 1–24, 2001.

[11] M. Harchol-Balter, "The Effect of Heavy-Tailed Job Size. Distributions on Computer System Design", In *Proc. of ASA-IMS Conf. on Applications of Heavy Tailed Distributions in Economics*, June 1999.

[12] M. Ilvesmaki and J. Karvo, "On the Behavior of the Candidate Table of the Per-flow Packet Count Flow Classifier", In *Proc. Eighth IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, volume 36, pp. 1–11, 2000.

[13] A. Khalil and T. Braun, "Egde Provisioning and Fairness in VPN-Diffserv Networks", In *ICCCN 2000*, pp. 424–431, October 2000.

[14] L. Kleinrock, *Queuing Systems, Volume II: Computer Applications*, Wiley, New York, 1976.

[15] H. Krayl, E. J. Neuhold, and C. Unger, *Grundlagen der Betriebssysteme*, Walter de Gruyter, Berlin, New York, 1975.

[16] A. Kumar et al., "Algorithms for Provisioning VPNs in the Hose Model", In *Proceedings of ACM SIGCOMM 2001*, pp. 135–146, 2001.

[17] H. Lee et al., "End-to-End QoS Architecture for VPNs: MPLS VPN Development in a Backbone Network", In *IEEE MILCOM 2000*, pp. 479–483, Los Angeles, USA, 2000.

[18] I. A. Rai, G. Urvoy-Keller, and E. W. Biersack, "On Reducing Response Time for VPN Traffic", TR 02.03.1, Institut Eurecom, March 2002.

[19] E. Rosen and Y. Rekhter, "BGP/MPLS VPN", Request for Comments RFC 2547, Internet Engineering Task Force, March 1999.

[20] L. E. Schrage, "A Proof of the Optimality of the Shortest Remaining Service Time Discipline", *Operations Research*, 16:670–690, 1968.

[21] P. Zhang and R. Kantola, "Building MPLS VPNs with QoS Routing Capability", In *Fifth International Symposium on Interworking*, pp. 292–301, October 2000.