# Performance Analysis of Stochastic Fair Sharing (SFS) Scheme for Link Sharing

R. Manivasakan, Mounir Hamdi and Danny H.K. Tsang

***Abstract*— We address the problem of the performance analysis of the Stochastic Fair Sharing (SFS) algorithm for fair link sharing. The SFS scheme has been proposed in [1] to carry out a fair link sharing and fair sharing among Virtual Private Networks (VPNs). Depending upon the current utilization and provisioned capacities of the classes, the SFS admission control algorithm decides which sessions to accept and which to reject. In this paper, we undertake the performance evaluation of the SFS scheme analytically. The main performance measure in our analysis is the session blocking probability. In particular, we obtain the Roberts-Kaufman's [2] like recursion for the SFS scheme to compute the blocking probability. We then use linear programming techniques to compute the blocking probability from the above recursion.**

**Keywords**: Link sharing, SFS, VPN

## I. INTRODUCTION

Link sharing schemes have been proposed to allow the service providers to lease a part of their physical link to independent organizations (through their Virtual Private Networks (VPNs) [3]). The complete sharing scheme delivers maximum possible BW (BW) usage efficiency while the complete partitioning scheme provides 'fairness'. In order to optimally use the BW (BW) capacity of the physical link and at the same time retain the fairness to the VPNs of varying session arrival rates, there were schemes [4], [1] proposed in literature which gave priority to the underloaded VPN's. In the Stochastic Fair Sharing (SFS) scheme proposed in [1], a certain amount of BW is reserved for a VPN of lower normalized BW usage before accepting a session belonging to a VPN of higher normalized BW usage. In SFS, the unused free capacity is *fairly* redistributed by resizing the capacity allocations depending upon the current usage of different VPNs sharing the link. On the contrary, in the scheme proposed in [4], the free capacity *cannot* be redistributed *fairly* over the overloaded classes using the technique of trunk reservation although the underloaded classes are given priority over the overloaded classes while accepting the sessions. For example, in this scheme, a high session arrival rate may take the residual capacity of all classes.

The problem of BW allocation to a VPN and a typical IP service is significantly different since the dynamism of these services are different with respect to their time scale of holding times. Typically, a VPN connection requires that a fixed BW is reserved for it for weeks or even months whereas a typical IP service has a holding time of just few minutes. This requires a redefinition of the notion of 'fairness' as defined by Parekh and Gallager [5]. This has been done in [1] where the notion of fairness is also extended to the concept of BW reservations.

In general, while studying the performance of systems where very complex models are encountered, simulation techniques are successfully employed and sometimes preferred to analysis due to the intractability of such models. However, there is a certain need for analytical results when possible to get deeper insight, reduce runtimes, handle very rare events and optimize system performance. Towards this end, we consider the analytical performance evaluation of SFS scheme which the work [1] lacks. In this work, we analytically derive the blocking probability (in terms of the parameters of the SFS scheme ) for sessions belonging to a class (VPN). It would analytically give a trade-off between fairness and efficiency of BW usage. The paper is organized as follows. The SFS scheme is explained briefly in Section 2. Section 3 discusses the analysis of the SFS scheme where we arrive at the global balance equation which explains the dynamics of the SFS scheme. From the global balance equation, we derive the Roberts-Kaufman's like recursion for the SFS scheme for computing the blocking probability. We then present the related simulation results. Conclusions are presented in Section 4.

This work was supported in part by the post-doctoral matching fund under Prof. Mounir Hamdi. Corresponding author: Prof. Mounir Hamdi email: hamdi@cs.ust.hk. Prof. Hamdi is with the Department of Computer Science, Hong Kong University of Science and Technology, Kowloon, Hong Kong. Dr. R. Manivasakan and Prof. Danny H. K. Tsang are with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong email: eemani@ee.ust.hk, eetsang@ust.hk

## II. The SFS scheme for link sharing

In this section, we describe the SFS scheme for the case of sharing in single link. For more details please refer to [1]. We consider a link of capacity $C$ to be partitioned into $N$ logical links (or classes) of provisioned capacities $C_i$, such that $\sum_{i=1}^{N} C_i \leq C$. We assume that real-time sessions arrive randomly according to a Poisson process. BW is reserved upon session arrivals and is released upon session completion. There is an admission control entity at the link which decides whether the link has adequate free capacity to accept the reservation requests of sessions. The session is said to be blocked if the session cannot be accepted.

Let $r_i$ be the amount of capacity currently used by a logical link $i$. The normalized usage of logical link $i$ is given by $u_i = r_i/C_i$. Consider the logical links being labeled in increasing order of their normalized usage. A new session of $i$th class is accepted only if the free capacity after accepting the session is greater than or equal to the sum of the trunk reservation with lower normalized usage. Mathematically, a new session of logical link $i$, with BW request $b_i$ is accepted if and only if, $\sum_{j=1}^{N} r_j + b_i + \sum_{j<i} t_j \leq C$ where $t_j$ is the *trunk reservation* for class $j$. The logical link with the lowest normalized usage is given the highest priority while accepting the sessions and hence sees a very low blocking probability. If the normalized usage $u_i$ of a logical link $i$ is close to its *fair share* denoted by $f_i$ (described below), then it is not necessary to have a large value of trunk reservation for the logical link. Hence, the trunk reservation $t_i$ is set to a static (maximum) trunk reservation parameter $\hat{t}_i$ when the difference between the fair share of a logical link and its current usage is large and is set to this difference if the difference is less than its static trunk reservation. Formally, $t_i = min[\hat{t}_i, f_i - r_i]$. The fair share of the logical link is the share it gets when the free capacity of logical links with lower normalized usage is shared by logical links with higher normalized usage. It is computed by redistributing the free capacity of logical links with lower normalized usage as follows: $f_i = \frac{C_i}{\sum_{j=i}^{N} C_j}\left(C - \sum_{j=1}^{i-1}(r_j + t_j)\right)$. The above expression is a natural generalization of the fairness criteria [5] used in packet schedulers.

We next use the state space diagram to get a deeper insight into the behavior of the SFS call admission algorithm. Consider a physical link being shared by two 'logical' links. The state of the (physical) link (as represented by any point in the state space diagram) at any

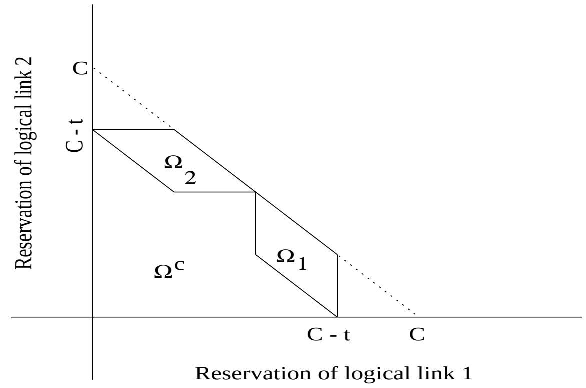


Fig. 1. State space diagram for a link

given time is represented by the current BW reservation of the two logical links. We assume that the trunk reservation and BW request for both the logical links are the same ($t_1 = t_2 = t,\ b_1 = b_2 = b$). Figure 1 shows the state space diagram of the system. The X and Y axes represent the current reservation of the first and second logical link respectively. The current state of the link can be represented by a point in the state space diagram. When a new session on the first logical link is accepted the system moves to the right while upon acceptance of a new session on the second logical link it moves up. When the sessions of the first or the second logical link complete, the system moves towards the left or downwards. As long as the total free capacity is greater than the trunk reservation, the system may move in any direction (assuming that the BW requirement of the sessions is small). This area is denoted by $\Omega^c$ in the figure. In the region where the available capacity is less than the trunk reservation (the area between the two slant lines), the session of the first logical link can be accepted only if the current reservation of the first logical link is less than that of the second logical link. Thus the system can move left, right or down but not upwards. This region is denoted by $\Omega_2$. In this region, class 2 session arrivals are prohibited. Similarly in the region denoted by $\Omega_1$ the system can move left, upwards or downwards but not towards the right. In this region, class 1 session arrivals are prohibited. We define for our convenience, $\Omega \triangleq \Omega^c \bigcup_l \Omega_l$. Naturally, $\Omega$ forms the set of all allowable states.

## III. Analysis of the SFS scheme

In this section, we evaluate the session blocking probability in a VPN. We first use the Markovian models to write first the global balance equation and then derive the Roberts-Kaufman's like recursion for the SFS case.

We then use linear optimization techniques to evaluate the session blocking probability.

*The Model, Assumptions and Notations:* We assume that sessions of class $k$ arrive according to a Poisson process with parameter $\lambda_k$ and have exponential holding times with mean $1/\mu_k$. The physical link capacity is $C$ units. We use $\mathbf{n}$ to denote the random vector $\mathbf{n} \triangleq (n_1, n_2, \ldots, n_K)$ where $n_i$ is the random variable denoting the number of type $i$ sessions using the physical link where $K$ is the number of classes of traffic handled by the physical link. Denote the stationary probability $P(n)$ of the system in state $\mathbf{n} = n$, i.e., $P(n) \triangleq Pr\{\mathbf{n} = n\}$. We use $b_k$ to denote the BW requirement for the session arrival of $k$th class and $\mathbf{b}$ to denote the vector $(b_1, b_2, \ldots, b_K)$. Define $\Gamma(i) \triangleq \{\mathbf{n} | \mathbf{n} \cdot \mathbf{b} = i\}$ where the notation $\mathbf{n} \cdot \mathbf{b}$ is used to denote the sum $\sum_{k=1}^{K} \mathbf{n}_k b_k$. Let $q(i) \triangleq Pr\{\mathbf{n} \cdot \mathbf{b} = i\}$. Finally, we assume a symmetrical system, i.e., $t_i = t$ for $i = 1, 2, \ldots, K$. We need the following notation $\mathbf{n}_i^+ = (n_1, \ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots, n_K)$, and $\mathbf{n}_i^- = (n_1, \ldots, n_{i-1}, n_i - 1, n_{i+1}, \ldots, n_K)$ Define the functions

$$\gamma_i^+(\mathbf{n}) = \begin{cases} 1 & \mathbf{n}_i^+ \in \Omega^c \bigcup_{l \neq i} \Omega_l \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_i^-(\mathbf{n}) = \begin{cases} 1 & \mathbf{n}_i^- \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad \alpha_i^+(\mathbf{n}) = \begin{cases} 1 & \mathbf{n}_i^+ \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

and $\alpha_i^-(\mathbf{n}) = \begin{cases} 1 & \mathbf{n}_i^- \in \Omega^c \bigcup_{l \neq i} \Omega_l \ \ and \\ & \mathbf{n} \in \Omega^c \bigcup_{l \neq i} \Omega_l \\ 0 & \text{otherwise} \end{cases}$

Let $E(\mathbf{n}) = \sum_{i=1}^{K} \lambda_i \gamma_i^+(\mathbf{n}) + \sum_{i=1}^{K} n_i \mu_i \gamma_i^-(\mathbf{n})$ and $H(\mathbf{n}) = \sum_{i=1}^{K} (n_i + 1) \mu_i \alpha_i^+(\mathbf{n}) P(\mathbf{n}_i^+) I_{\{\sum_k \mathbf{n}_k b_k \leq C - b_i\}}$. One can write the global balance equations for SFS using the definitions above as,

$$E(\mathbf{n}) P(n) = \sum_{i=1}^{K} \lambda_i \alpha_i^-(\mathbf{n}) P(\mathbf{n}_i^-) \ \ I_{\{n_i > 0\}} + H(\mathbf{n}) \quad (1)$$

where $I_{\{.\}}$ is the indicator function. It is worth noting that the link occupancy constitutes an irreducible Markov process with the feasible region $\Omega$ as the state space. Denote the blocking probability for class $i$ calls by $\beta_i$. We assume that $\mu_l = \mu$ and $b_l = 1$ for $l = 1, 2, \ldots, K$ in order to make the analysis feasible.

*Roberts-Kaufman's Type Recursion for SFS:* In this subsection, we obtain the Roberts-Kaufman's like recursion for SFS [2]. It is to be noted that the number of connections of type $i$ at any time form a *dependent* Birth-Death (BD) chain in SFS. (This is in sharp contrast with the coordinate convex policies, where the local balance equation is always satisfied). Hence, the Roberts-Kaufman's recursion for SFS is quite difficult to obtain. But, under the assumptions mentioned in the previous section, one can obtain a similar recursion which we will derive now. We start with the definition,

$$B_l(i) = \begin{cases} \sum_{j=t}^{C-t} D_l(i,j) & i = C \\ \sum_{j=(int)(C/2.0)}^{C-t} D_l(i,j) & C - t \leq i < C \\ 0 & otherwise. \end{cases}$$

where $D_l(i,j) = Pr\{W = i, n_l = j\}$. Then, the blocking probability $\beta_k$ for a session belonging to the class $k$ is given by $\beta_k = \left(\sum_{i=C-t}^{C} B_k(i)\right) / \left(\sum_{i=0}^{C} q(i)\right)$. Now, we give a recursion in $q(i)$ and $B_l(i)$ such that $B_l(i)$ can be computed. This recursion is similar to Roberts-Kaufman's recursion. It can be shown that the summing over $\{\mathbf{n} | \mathbf{n} \in \Gamma(i)\}$ on either side of (1) and after some manipulations gives,

$$\sum_{l=1}^{K} \lambda_l \left[q(i) - B_l(i)\right] + \mu \sum_{l=1}^{K} \sum_{\mathbf{n} \in \Gamma(i)} n_l P(n) =$$

$$\sum_{l=1}^{K} \lambda_l \left[q(i - b_l) - B_k(i - b_l)\right] + \mu \sum_{l=1}^{K} \sum_{\mathbf{k} \in \Gamma(i + b_l)} k_l P(k) \quad (2)$$

Consider, $iq(i) - (i+1)q(i+1) = b \sum_{l=1}^{K} \sum_{\mathbf{n} \in \Gamma(i)} n_l P(n) - b \sum_{l=1}^{K} \sum_{\mathbf{n} \in \Gamma(i+1)} n_l P(n)$ where, we have made use of the assumption $b_l = 1$ for $1 \leq l \leq K$. Using (2) we have,

$$(i+1)q(i+1) = iq(i) + 1/\mu \sum_{l=1}^{K} \lambda_l \left[q(i) - B_l(i)\right] -$$

$$1/\mu \sum_{l=1}^{K} \lambda_l \left[q(i-1) - B_l(i-1)\right] \quad (3)$$

for $0 \leq i \leq C - 1$. The above equation gives the recursive computation of $q(i)$ and $B_l(i)$. Note that, for the SFS scheme, the $q(i+1)$ depends not only on $q(i)$ and $q(i-1)$, but, also on $B_l(i)$ and $B_l(i-1)$. It is the presence of $B_l(i)$'s which prevent the usefulness of the recursive relationship. However, we identified that linear optimization techniques can be used to compute the blocking probabilities $\beta_k$. In what follows, we arrive at a linear optimization problem from the above recursive equation.

We start with the recursive Equation (3) and evaluate it for $i = C - 1$ which is given by,

$$Cq(C) \ = \ (C - 1 + Q)q(C-1) - Qq(C-2)$$

$$-S(C-1)+S(C-2) \quad (4)$$

where, $Q=\frac{1}{\mu}\sum_{i=1}^{K}\lambda_i$ and $S(i)=\sum_{l=1}^{K}\lambda_l B_l(i)$.

Using the linear equations (for $i=1$ through $i=C-2$) from (3), one can deduce the following relationship,

$$\prod_{i=1}^{C-3}\frac{(C-i-1)}{D_{C-4}}q(C-2)=\frac{D_{C-3}}{D_{C-4}}q(1)-Qq(0)-\prod_{l=1}^{C-4}\frac{C-2-l}{D_{C-4}}S(C-3)-\frac{Q}{D_{C-4}}\prod_{l=2}^{C-4}(C-2-l)S(C-4)+S(0)+\sum_{j=5}^{C-3}\left[\frac{(C-1-j)D_{j-2}-D_{j-1}}{D_{C-4}}\right]\times\prod_{l=i}^{C-4}(C-l-2)S(C-j-2) \quad (5)$$

Here, the sequence $D_k$ for $k>2$ is given by, $D_k=(R-k+1+Q)D_{k-1}-Q(R-k+2)D_{k-2}$ where $D_1=R+Q$, $D_2=(R-1+Q)(R+Q)-QR$ and $R=C-3$. Note that $S(i)=0$ for $i<C-t$. Substituting the value of $q(C-2)$ from (5) in (4), we have,

$$Cq(C)=(C-1+Q)q(C-1)-S(C-1)-\frac{D_{C-3}}{\prod_{i=1}^{C-3}(C-i-1)}q(1)Q-\frac{QS(0)D_{C-4}}{\prod_{i=1}^{C-3}(C-i-1)}+\frac{D_{C-4}Q^2q(0)}{\prod_{i=1}^{C-3}(C-i-1)}+\frac{\prod_{l=1}^{C-4}(C-2-l)}{\prod_{i=1}^{C-3}(C-i-1)}QS(C-3)+Q^2S(C-4)\frac{\prod_{l=2}^{C-4}(C-2-l)}{\prod_{i=1}^{C-3}(C-i-1)}+S(C-2)-\frac{QD_{C-4}}{\prod_{i=1}^{C-3}(C-i-1)}\sum_{j=5}^{C-3}\left[\frac{(C-1-j)D_{j-2}-D_{j-1}}{D_{C-4}}\right]\times\prod_{l=j}^{C-4}(C-l-2)S(C-j-2) \quad (6)$$

Expressing $q(C-1)$ in terms of $q(1)$ and $q(0)$ using the linear Equations (3) for $i=1$ through $i=C-1$, we have,

$$\frac{1}{D_{C-3}}\prod_{i=1}^{C-2}(C-i)q(C-1)=q(1)\frac{D_{C-2}}{D_{C-3}}-Qq(0)-\frac{\prod_{l=1}^{C-3}(C-l-1)S(C-2)}{D_{C-3}}-\frac{Q}{D_{C-3}}\prod_{l=2}^{C-3}(C-1-l)S(C-3)+\sum_{j=4}^{C-2}\left[\frac{(C-j)D_{j-2}-D_{j-1}}{D_{C-3}}\right]\times\prod_{l=j}^{C-3}(C-l-1)S(C-j-1)+S(0) \quad (7)$$

Now, consider the following identity, $q(C)+q(C-1)+\sum_{j=2}^{C-2}q(j)+q(1)+q(0)=1$ where $q(j)$ can again be written in terms of $q(1)$ and $q(0)$ using recursive relations given by (3) (exploiting only equations with index $i$ going from $j$ to 1). Now, we have two linear equations involving $q(1)$ and $q(0)$, namely Equation (7) and the above identity which can be solved for $q(1)$ and $q(0)$. Thus obtained values are substituted in (6) to obtain the following equation,

$$0=(C-1+Q)q(C-1)-\frac{D_{C-3}}{\prod_{i=1}^{C-3}(C-i-1)}q(1)Q+\frac{D_{C-4}Q^2q(0)}{\prod_{i=1}^{C-3}(C-i-1)}+\frac{\prod_{l=1}^{C-4}(C-2-l)}{\prod_{i=1}^{C-3}(C-i-1)}QS(C-3)+Q^2S(C-4)\frac{\prod_{l=2}^{C-4}(C-2-l)}{\prod_{i=1}^{C-3}(C-i-1)}-\frac{QD_{C-4}}{\prod_{i=1}^{C-3}(C-i-1)}\times\sum_{j=5}^{C-3}\left[\frac{(C-1-j)D_{j-2}-D_{j-1}}{D_{C-4}}\right]\times\prod_{l=j}^{C-4}(C-l-2)S(C-j-2)-\frac{QS(0)D_{C-4}}{\prod_{i=1}^{C-3}(C-i-1)}-S(C-1)+S(C-2)-Cq(C) \quad (8)$$

where, $q(1)=\frac{QP+M\left[1-Q\sum_{j=C-2}^{2}\frac{D_{j-2}}{\prod_{l=1}^{j-1}(j-l+1)}\right]}{DNMTR}$ $q(0)=\frac{P\frac{D_{C-2}}{D_{C-3}}-M\left[1+\sum_{j=C-2}^{2}\frac{D_{j-1}}{\prod_{l=1}^{j-1}(j-l+1)}\right]}{DNMTR}$ where again,

$$P\triangleq 1-q(C)-q(C-1)-\sum_{j-2}^{2}\frac{D_{j-2}}{\prod_{l=1}^{j-1}(j-l+1)}\times\left[S(0)-\frac{S(j-1)\prod_{l=1}^{j-2}(j-l)}{D_{j-2}}-\frac{QS(j-2)}{D_{j-2}}\prod_{l=2}^{j-2}(j-l)+\sum_{l=3}^{j-1}\left[\frac{(j-l+1)D_{l-2}-D_{l-1}}{D_{j-2}}\right]S(j-l)\right] \quad (9)$$

$$M\triangleq\frac{\prod_{i=1}^{C-2}(C-i)q(C-1)}{D_{C-3}}+\frac{\prod_{l=1}^{C-3}(C-l-1)}{D_{C-3}}S(C-2)+\frac{Q\prod_{l=2}^{C-3}(C-l-1)}{D_{C-3}}S(C-3)-S(0)-\sum_{j=4}^{C-2}\left[\frac{(C-j)D_{j-2}-D_{j-1}}{D_{C-3}}\right]\prod_{l=j}^{C-3}(C-l-1)S(C-j-1) \quad (10)$$

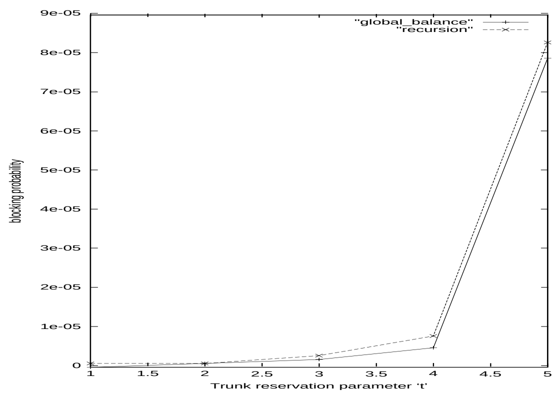


Fig. 2. Blocking Probability for Type 1 Class

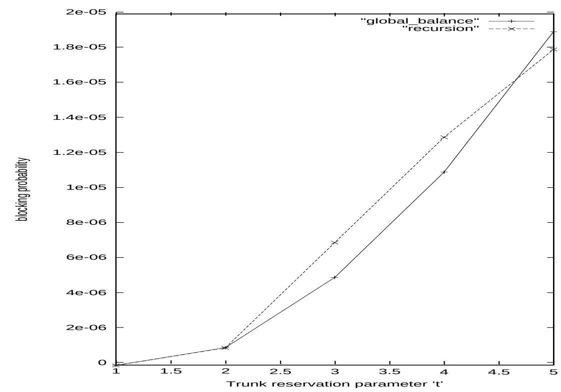


Fig. 3. Blocking Probability for Type 2 Class

$$DNMTR \triangleq Q\left[1+\sum_{j=C-2}^{2}\frac{D_{j-1}}{\prod_{l=1}^{j-1}(j-l+1)}\right]+ \frac{D_{C-2}}{D_{C-3}}\left[1-Q\sum_{j=C-2}^{2}\frac{D_{j-2}}{\prod_{l=1}^{j-1}(j-l+1)}\right] \quad (11)$$

Now, note that Equation (8) is a function of $q(C),\ q(C-1)$ and $B_l(i)$ for $1 \le l \le K$ and $C-t \le i \le C$. One can now formulate the above problem of finding the variables $q(C-1)$ and $B_l(i)$ into an optimization problem [6] with the RHS of the equation (8) as the cost function to be minimized with the following constraints, (i) $q(C) = B_l(C)\ \ 1 \le l \le K$ (ii)$0 < q(C-1) < 1$ (iii) $0 < B_l(i) < 1$ for $1 \le l \le K$ and $C-t \le i \le C$ and (iv) $q(C-1) > B_l(C-1)$ for $1 \le l \le K$.

### A. Simulation Results:

We carry out experiments, to check the efficacy of the recursive equation (3) in terms of computational savings. For our experiments, we have taken $\lambda_1 = 3.0$, $\lambda_2 = 3.4$, $b_1 = b_2 = 1$ and $\mu_1 = \mu_2 = \mu = 4.0$. Also the capacity of the link is taken to be $C = 12$. We first computed the blocking probability using the global balance equation (1). Then, we computed the blocking probability from the recursive equation (3) (by solving the corresponding optimization problem). As one can see from the Figures 2 and 3, the blocking probability obtained from the global balance equation (1) and the blocking probability from the recursive equation (3) (computed using optimization techniques) agree very well. But, the computational advantage we gain is around 50 % from the experimental observation.

## IV. CONCLUSIONS:-

In this paper, we studied the analytical performance evaluation of stochastic fair sharing (SFS) scheme to carry out fair link sharing. We developed a fairly accurate model based on the combination of different techniques, namely Markovian models, Roberts-Kaufman recursion and linear optimization. The main performance measure is the session blocking probability. We found that there was a good match between the blocking probability computed from the global balance equation and the blocking probability estimated from the recursive equation derived in the paper while the computational savings incurred were around 50 %. For reasonably large system (C large and number of classes considered large as against only two considered here), estimating the rare session blocking probability by simulation is inefficient and sometimes impossible. Our work is significant in this context that the recursive equation can be used in practice to evaluate the session blocking probability resulting in significant computational savings.

## REFERENCES

[1] Rahul Garg and Huzur Saran, ""Fair bandwidth sharing among virtual networks: a capacity resizing approach"," *Infocom*, 2000.

[2] J. S. Kaufman, ""Blocking in a Shared Resource Environment "," *IEEE Trans. commun.*, vol. COM-29, no. 19, pp. 1474–1481, Oct 1981.

[3] B. Gleeson, A. Lin, J. Heinannen, G. Armitage and A. Malis, ""A Framework for IP Based Virtual Private Networks"," *Internet Draft*, Sep. 1998.

[4] S. Borst and D. Mitra, ""Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic"," *IEEE J. Select. Reas in Commun.*, vol. 16, no. 5, pp. 668–678, June 1998.

[5] A. K. Parekh and Robert G. Gallager, ""A generalised processor sharing approach to flow control- the single node case"," *Proceedings of INFOCOM*, vol. 16, pp. 915–924, May 1992.

[6] B. Kolman and R. E. Beck, *"Elementary Linear Programming with Applications"*, Academic Press, 1980.