A Simple IP Flow Blocking Model

Alexander A. Kist[a], Bill Lloyd-Smith[a] and Richard J. Harris[b]

[a]Centre for Advanced Technology in Telecommunications and
School of Computer Science and Information Technology
RMIT University
BOX 2476V, Victoria 3001, Australia
Email: kist@ieee.org and cwls@cs.rmit.edu.au

[b]Institute of Information Sciences and Technology
Massey University
Private Bag 11 222, Palmerston North, New Zealand
Email: R.Harris@massey.ac.nz

Abstract: Flow based networking can assist in addressing current performance issues in convergent IP networks and enable dynamic, real-time routing. The Scheme for Advanced Overflow Routing (SAPOR) is a method that enables flow based routing in IP networks. This paper presents an analytical model that allows the calculation of overflow probabilities for flows, transmitted via a bottleneck connection. The proposed model is verified by simulation and results are presented. Similar to the Erlang B formula, this proposed model can also be used as a generic flow blocking model, i.e. when call rates (required resources) are drawn from probability distributions.

Key-Words: Routing, Blocking Probability, Flow Based Networking, Overflow Routing, SAPOR

1. INTRODUCTION

Most routing regimes in IP networks are not sensitive to network loads and are static in between route changes. Routing protocols select a number of routes and these are used until new routes are calculated. Usually, this occurs when topologies change. Such behaviour is not always favourable and might have an impact on performance. Resulting problems have been acknowledged by the research community for a long time: In the case of incorrectly dimensioned resources, changes in network traffic or equipment failures, certain links in the network may become congested whereas other network parts may be underutilised.

Methods that allow traffic engineering and load distribution in IP networks have been proposed to address this problem. *Multiprotocol Label Switching* (MPLS) [1], for example, introduces a connection oriented model to IP environments, and separates data and control plane functions. MPLS allows traffic engineering by defined traffic routes. Other work proposes load distribution by *Open Shortest Path First* (OSPF) weight optimisation. OSPF is used to spread network load more evenly and enhance the network's ability to cope with route failures (e.g. [2], [3]). These methods rely heavily on accurate knowledge of traffic demands and none of these mentioned methods allow load distribution on the fly in real-time.

In *Public Switched Telephone Networks* (PSTNs) dynamic routing schemes have been used for a long time, examples include Dynamic non-hierarchical routing (DNHR) [4] which uses different path sets for different times of the day, *Dynamically Controlled Routing* (DCR) [5], *Dynamic Alternative Routing* (DAR) [6] and *State- and Time-Dependent Routing* (STR) [7]. Not many similar, dynamic schemes have been proposed for IP networks.

The prerequisite for realistic dynamic routing strategies in packet switched networks is the possibility of flow based routing or, in other words, a flow switched network. In the past, flow based routing has not received much attention, perhaps since it has been seen as non scalable. Changes in technology in the last decade make it possible to overcome some of these limitations. *Caspian Networks*, a start-up business, for example, promotes flow-based routers [8] and claims that their equipment can handle 6 million flows per $10G$ interface. The general notion, as such, has not received much attention; however the concept of flows and microflows is widely used. The *Equal-Cost Multipath* (ECMP) [9] mechanism, for example, is used by OSPF, utilising flow information to split larger flow aggregates across alternative interfaces.

Earlier work proposed the *Scheme for Alternative Packet Overflow Routing* (SAPOR) [10]. SAPOR enables overflow routing in IP networks and routes traffic on a flow bases. To be able to judge the performance of overflow schemes, to dimension resources and evaluate the operation of dynamic schemes, mathematical models are required that allow the calculation of flow blocking probabilities. The well-known Erlang B formula does not provide accurate results if the flows have different rates. Kaufman [11] and Roberts [12] independently proposed a model that allows the calculation of blocking probabilities for distinct flow classes. Several authors suggest the use of the Kaufman/Roberts method for continuously distributed flow rates by dividing the arrivals in discrete classes before applying the Kaufman and Roberts method. This process requires the knowledge of the flow rate distribution. This paper proposes a simple model which only requires the knowledge of mean and variance of the flow rate distribution.

The contributions of this paper are threefold: Firstly, practical performance parameters for flow based networking are defined; secondly, a mathematical model is presented which allows blocking probability calculations for flows; and thirdly, network flow simulation results are given for the proposed model. The paper is organised as follows: Section 2 introduces the SAPOR scheme to show the reader the practical background for the model, and Section 3 discusses performance parameters for flow based networking. The model is developed in Section 4 and the simulation results are given in Section 5. The paper concludes with remarks on the problem.

2. SAPOR

The general aim of the SAPOR research effort is to allow for automatic load distribution in the case of overloaded paths. This can be achieved by the use of the simple networking paradigm of flow-based routing which enables reliable and efficient routing strategies in IP networks. This section introduces the SAPOR scheme, to give the reader a background of the practical application of the model, introduced in Section 4.

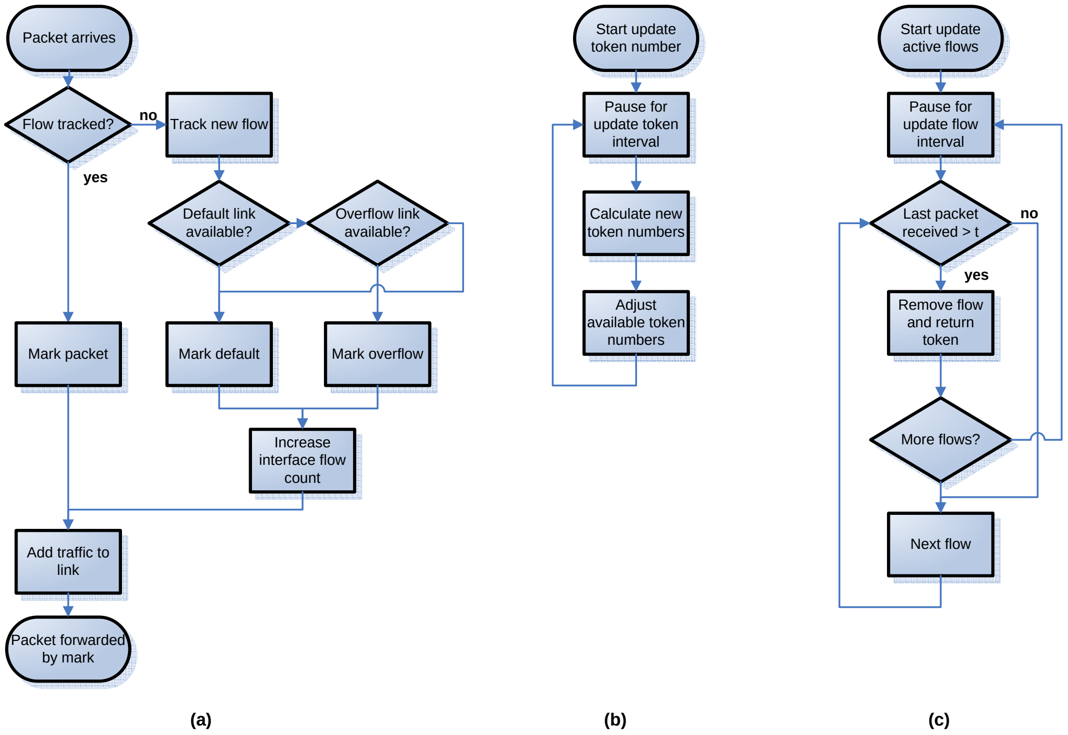

Figure 1. SAPOR Scheme - Routing

The SAPOR scheme enables flow based routing and implements three principles: Firstly, it ensures that packets that belong to the same microflow are routed on the same interface, even in the overflow case. Secondly, it determines how many additional microflows can be accommodated by the default link before it's target bandwidth is reached. And thirdly, if the target bandwidth is reached, additional flows are routed on alternative interfaces. A hash based flow tracker implements the first principle; the second and third principles are implemented by a token system.

Figure 1 (a) depicts a flow chart of the SAPOR operation. When a packet arrives, it is determined if the packet is part of a flow, already tracked. If the packet belongs to an existing flow, the packet is marked for transmission on a specific interface. Then the traffic is added to the link traffic measure and the packet is forwarded on the basis of the interface mark.

If the flow is not yet tracked, a new flow is added. It is determined if the flow can be accommodated by the default interface. If capacity is available, the flow is routed on the default link, the packet is marked and the interface flow count is increased. As before, the traffic is added to the link and the packet is forwarded. If the default link is not available, the availability of overflow links is determined. If there are no vacant overflow links, the flow is routed on the default link (or is dropped). Otherwise the flow is routed on the overflow link and the same steps are executed: the interface flow count is increased, the traffic is added and the packet is forwarded.

Tracked flows have to be cleared after they are finished. This is done by clearing inactive flows (Figure 1 (c)). If no more packets are received within a certain time interval (eg. 1 sec) belonging to the same flow, the tracker is cleared and the corresponding token is returned to the link budget. The available capacity on the links is determined by a token system. The number of tokens per link is calculated by the average flow size and the remaining bandwidth. At set time intervals it is updated to adapt for changes in the current average flow size (Figure 1 (b)). There are several possible implementations of this system, in the simplest, one token accounts for one flow, although more elaborate systems are possible.

The main reservations about SAPOR concern the scalability of the approach. Processing and memory requirements for each flow are minimal. More complex tasks are executed in longer time intervals; therefore, the main parameter with an impact on scalability is the number of flows that have to be tracked. Caspian networks quote peaks of 80,000 active flows for a traffic volume of 350 Mbps and average flow durations of 10 seconds. This number can be interpolated for larger capacities, but it depends heavily on the micro flow definition. Flows consist of related information and for performance reasons it is necessary that packets that belong to the same flow are routed on the same link. In this case, the maximum inter-arrival time can be relatively short, for instance 1 second long. If packets arrive with more than 1 second inter arrival time, the jitter is not a critical parameter and such packets can be routed on different paths, thus treated like different flows. Further investigations into scalability are currently being undertaken, including a prototype implementation.

3. PERFORMANCE MEASURE AND FLOW BASED NETWORKING

If the concept of flow based routing is used, parameters are required to judge the performance of such schemes. In circuit switched telephony networks as well as in QoS routing and admission control schemes, blocking probability or Grade of Service (GoS) are used as performance measures. For packet based networks accommodating elastic flows, GoS is not defined. IP network performance parameters are delay, packet loss and jitter. To judge the performance of flow based routing schemes another meaningful measure is required.

For the following discussion it is assumed that the current network is not the principal bottleneck for TCP flows and flows are assumed to have a constant, fixed rate over time. Furthermore, it is assumed that as long as the link utilisation is below a target utilisation, performance parameters are within acceptable boundaries. This bandwidth is referred to as target bandwidth γ: $\gamma = C \cdot u$. The target bandwidth is the capacity C multiplied by the target utilisation u. Based on these observations, it is possible to define two measures: The percentage of flows that are above the target bandwidth and the sum of traffic above the target bandwidth. A flow based GoS parameters for flow based networks (GSF) is defined by:

$$GSF_f = \frac{I - \max_{1 \le i \le I} \left\{ \sum_{j=1}^{i} 1 \mid \sum_{j=1}^{i} \varepsilon_j \le \gamma \right\}}{I} \tag{1}$$

where I denotes the number of offered flows and ε_i denotes the rate of flow number i.

These flows are sorted by their arrival time by increasing i. GSF_f is the proportion of flows above the target bandwidth. The second traffic based definition, GSF_t, is given by:

$$GSF_t = \frac{\sum_{j=1}^{I} \varepsilon_j - \max_{1 \leq i \leq I} \left\{ \sum_{j=1}^{i} \varepsilon_j \mid \sum_{j=1}^{i} \varepsilon_j \leq \gamma \right\}}{\sum_{j=1}^{I} \varepsilon_j} \tag{2}$$

GSF_t denotes the proportion of traffic which is above the target bandwidth. In the reminder of this paper, the GSF_f is used and referred to as flow blocking probability. The next section introduces a model that allows the calculation of this probability.

4. OVERFLOW MODEL

The main purpose of SAPOR is to enable overflow routing. This section introduces a mathematical model that allows the calculation of overflow probabilities in flow based environments for statistical equilibrium. Flows can be specified by their arrival time, their duration (service rate) and their rate (bandwidth requirement). For the calculations in this section it is assumed that flow arrivals follow a Poisson process[1] and it is assumed that the service rate is exponentially distributed[2]. Therefore, similar to the Erlang formula, the determining parameter is the traffic A where $A = \lambda \cdot d$ for a flow arrival rate of λ. The flow rate is assumed to be a random variable drawn from a distribution with mean μ_f and standard deviation σ_f. Links of capacity C have a usable bandwidth γ, defined by the maximum allowed utilisation u.

If the standard deviation of the flow rate is small and therefore all flows have similar rates, the overflow probability can be estimated with the well known Erlang B formula:

$$E(\varepsilon, A) = \frac{\frac{A^\varepsilon}{\varepsilon!}}{\sum_{i=0}^{\varepsilon} \frac{A^i}{i!}} \tag{3}$$

where $\varepsilon = \lfloor \frac{\gamma}{\mu_f} \rfloor$ is the number of flows which can be accommodated on the link. If flow rates are not equal, the Erlang formula does not approximate the blocking correctly. In this case it is determined by two factors: The flow arrival rate and the flow rate distribution.

This section outlines the implications of both factors. Firstly, the impact of the flow arrival and traffic volume are discussed; secondly, the influence of the flow rate distribution is considered and finally, the appropriate combination of both calculations is outlined.

The number of active flows is determined by the arrival assumption. For the Poisson process the probabilities that i flows are active, can be readily evaluated by:

$$P_p(i, A) = \frac{A^i}{i!} e^{-A} \tag{4}$$

[1] There has been much discussion addressing the Poisson assumption in packet based networks and concerning Internet flows. There are also indications that for a large number traffic volumes the arrival process approaches a Poisson process [13].

[2] As for Erlang, this restriction is not necessary. The loss formula are also valid for general holding times [14].

The sum $P_{ps}(i, A)$ of the first i arrivals is shown in Equation (5). This sum is required for the calculations in Equation (12)[3].

$$P_{ps}(i, A) = \sum_{j=0}^{i} \frac{A^j}{j!} e^{-A} \tag{5}$$

These equations govern the number of arriving flows; the influence of the flow rate is discussed next. The mean flow rate determines the number of flows that can be accommodated by γ. The process of selecting i flows from flow rate distribution can be described by drawing a number of i samples from a probability distribution with mean μ_f and standard deviation σ_f. To get the sampling distribution of the mean, the central limit theorem can be applied: The sample mean $\overline{y}$ is approximately normally distributed with mean $\mu_{\overline{y}} = \mu_f$ and standard deviation of $\sigma_{\overline{y}} = \frac{\sigma_f}{\sqrt{i}}$.

The maximum possible number of flows that can be accommodated by a link can be modelled by drawing i samples from this probability distribution until $\sum_{j=0}^{i} y_i \leq \gamma$ and $\sum_{j=0}^{i+1} y_i > \gamma$, where y_i is the flow rate sample. This describes the situation where i flows can be accommodated, but $i+1$ cannot. The probability $P_n(i, \mu_f, \sigma_f)$ that the average flow rate is between these threshold can be calculated by:

$$P_n(i, \mu_f, \sigma_f) = P\left(\frac{\gamma}{i} \leq \overline{y} < \frac{\gamma}{i+1}\right) \tag{6}$$

This is the probability that exactly i flows can be accommodated by γ. Since $\overline{y}$ is normally distributed the z values z^- and z^+ can be calculated by:

$$z^- = \frac{\frac{\gamma}{\sqrt{i}} - \mu_f\sqrt{i}}{\sigma_f} \tag{7}$$

$$z^+ = \frac{\frac{\gamma}{\sqrt{i+1}} - \mu_f\sqrt{i+1}}{\sigma_f} \tag{8}$$

and the probability $P_n(i, \mu_f, \sigma_f)$ is given by:

$$P_n(i, \mu_f, \sigma_f) = P(z^- \leq z < z^+) \tag{9}$$

Equation (9) can be evaluated by standard methods for normal distributions and $P_n(i, \mu_f, \sigma_f)$ is calculated. In the remainder of this section the combination of the two models is discussed.

Figure 2 depicts three examples to outline the situation. The graphs show probabilities verses virtual slot numbers for a Poisson distribution for $A = 10$ and three different normal-model (Equation 9) distributions[4]: Fig. 2(a) $\sigma = 0$, Fig. 2 (b) $\sigma = 0.1 \cdot \mu$ and Fig. 2 (c) $\sigma = \mu$. It can be observed that for $\sigma = 0$ only $i = 10$ flows can fill the bandwidth. For $\sigma = 0.1 \cdot \mu$, six different flow numbers are likely ($P > 99\%$): $i = 8, \ldots, 13$ and for

[3] Note that the Erlang blocking can be calculated by: $E(i, A) = \frac{P_p(i,A)}{P_{ps}(i,A)}$.

[4] The distributions for the normal-model in Figure 2 are skewed since there is a dependency of $P_n(i, \mu_f, \sigma_f)$ on the number of samples i.

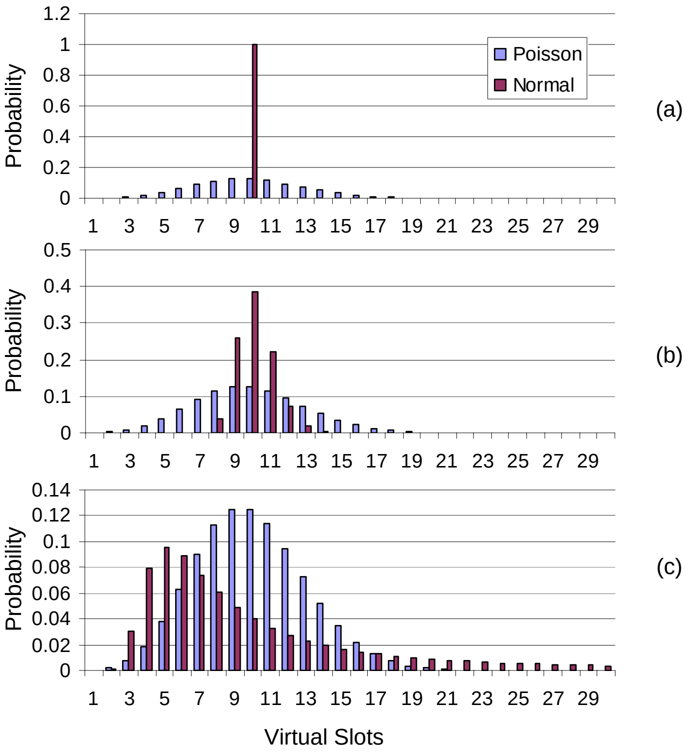

Figure 2. Example: Poisson and Normal Distributions

$\sigma = \mu$ a higher spread of flow numbers is likely. The combined blocking probability and therefore the event that i flows arrive and i flows fill the available bandwidth is given by:

$$P_{pn}(i, A) = P_p(i, A) \cdot P_n(i, \mu_f, \sigma_f) \tag{10}$$

Equation 10 gives the probability that the bandwidth limitation is reached by i flows and that i flows arrive. To get the overall blocking $P(A)$ all possible cases have to be summarised, viz:

$$P(A) = \sum_j P_p(j, A) \cdot P_n(j, \mu_f, \sigma_f) \tag{11}$$

However, Equation (11) does not consider that some flows will be blocked. As for Erlang blocking, the truncated case has to be considered. To calculate the truncation, it is required to know the maximum number of flows (or their distribution) possible on the link. This number is unknown, but the best estimate is the maximum occupied slot numbers follow before calculated normal distribution. Using this assumption, Equation (11) can be normalised by the sum of accumulated Poisson arrivals (Equation (5)) multiplied by the probability that i slots are active (Equation (9)), viz:

$$P(A) = \frac{\sum_j P_p(j, A) \cdot P_n(j, \mu_f, \sigma_f)}{\sum_j P_{ps}(j, A) \cdot P_n(j, \mu_f, \sigma_f)} \tag{12}$$

This calculation follows the pattern of the Erlang formula. Depending on proportions of the normal distribution, blocking probabilities are selected from Poisson distribution. It is normalised by the sum of the corresponding values for truncated cases. For small $\sigma_{\overline{y}}$, all but one $P_n(i, \mu_f, \sigma_f)$ approach zero and Equation (12) approaches the Erlang blocking formula.

Note: the sums for probability calculations run from 0 to infinity. For practical calculations however a threshold can be used to truncate the sums without recognisable changes in results. These formulas can easily be implemented by program code. The next section presents the analysis of the model.

5. SIMULATION RESULTS

To verify the analytical results, a discrete event flow simulator was used which utilises the Mersenne Twister [15] random number generator with a period of $2^{19937-1}$. The simulator was executed for two simulated hours and the warm-up period was also two hours. The blocking was measured every second and therefore 7200 measurements were available per simulation run. The result graphs depict 95% confidence intervals for these measurements. During different simulation-runs the available capacity was varied and the flow arrival rate was kept constant. The graphs show the capacity in units of average capacity (virtual slots), for example, $1,500,000$ bytes are equivalent to 500 slots for an average flow rate of 3000 bytes. The average flow duration was set to $d = 10s$ and average flow rate was set to 3000 bytes/sec.

The first set of simulations used a flow arrival rate of $\lambda = 50\ \frac{1}{s}$, and the flow rate standard deviation, as well as the available capacity were varied: $1478, 3000, 6000$ and between $12 \cdot 10^6$ and $18 \cdot 10^6$ bytes (400 and 600 slots) respectively. The lines in the graph show the analytical results for Erlang Loss, the model with $\frac{\sigma_f}{\mu_f} = 0.558$, $\frac{\sigma_f}{\mu_f} = 1$ and $\frac{\sigma_f}{\mu_f} = 2$. The error bars depict 95% confidence intervals of the simulation results. Figure 3 depicts the blocking probability versus the available capacity. The capacity is shown as multiples of the average flow rate μ_f. Figure 4 depicts the same results with a logarithmic scale to show the details for higher blocking probabilities. The offered traffic is equivalent to 500 slots. If the capacity is decreased the blocking increases, for higher available capacities blocking is reduced. The model agrees with simulation in most cases. The only notable exceptions are the cases where the available capacity is below the average load and the $\frac{\sigma}{\mu}$ ratio is increased. For higher capacities blocking becomes a rare event and the enlarged confidence intervals can be noticed.

A second set of simulations is depicted in Figure 5 and Figure 6. The flow arrival rate was $\lambda = 5\frac{1}{s}$ and the flow rate standard deviation was varied: $1478, 3000, 6000$. The available capacity was changed between $75,000$ and $225,000$ (25 and 75 slots). The offered traffic was equivalent to 50 slots. As before, the model shows good agreement, in particular, for capacities above the average load. For increased $\frac{\sigma}{\mu}$ ratios and less available bandwidth the discrepancies between model and simulation occur. This effect is more pronounced than in the first simulation-set with the 500 virtual slots load. Furthermore, the results for the case $\frac{\sigma}{\mu} = 2$ ratio fluctuate a lot. In this case only a small number of different flows are sampled from a distribution with high variance. This results in high fluctuations.

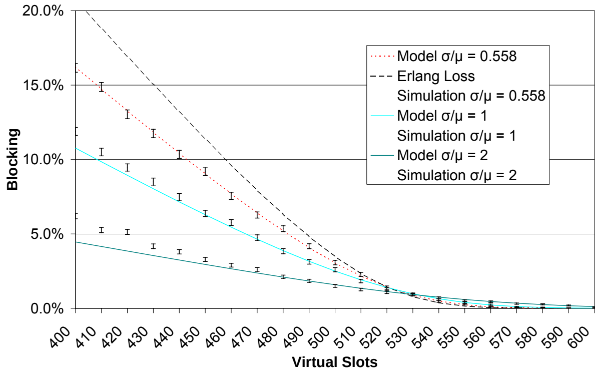

Figure 3. Blocking versus Capacity $[\mu_f]$

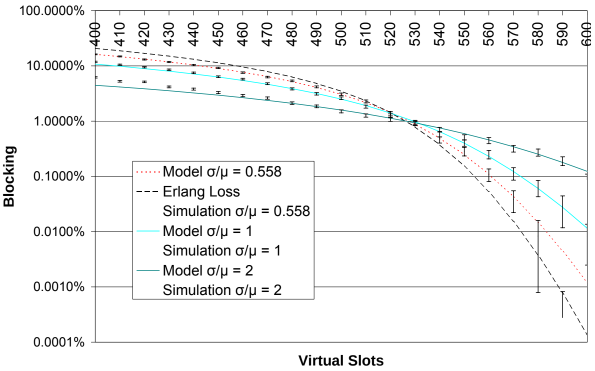

Figure 4. Logarithmic Bocking versus Capacity $[\mu_f]$

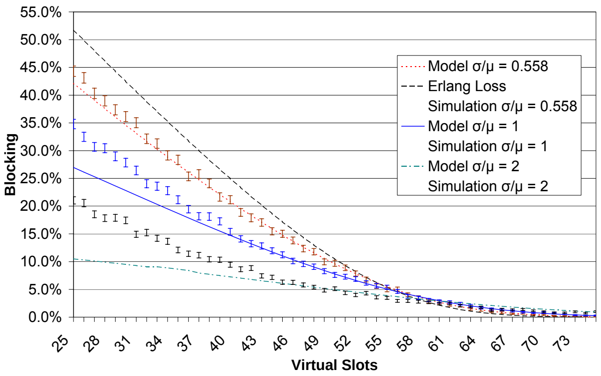

Figure 5. Blocking versus Capacity $[\mu_f]$

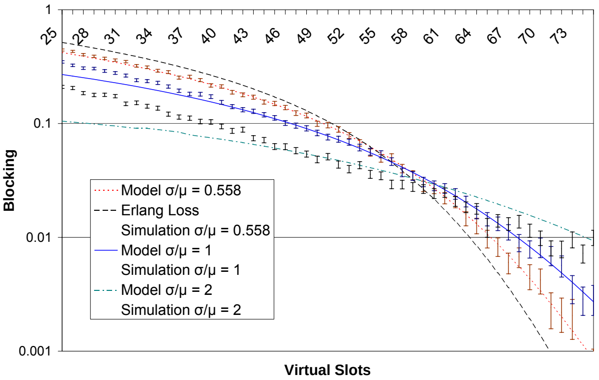

Figure 6. Logarithmic Bocking versus Capacity $[\mu_f]$

6. CONCLUSIONS

This paper proposed a model that allows the blocking probability calculation in the case where flows have different rates with known mean and standard deviation. Simulation results were presented to verify the model. For situations where the average offered traffic is at or below the available bandwidth, the model agrees well with the simulation, for high blocking probabilities and larger σ/μ ratios, the model underestimates blocking. This can be explained: If the average offered traffic is higher than the available bandwidth, larger flows are more likely to be blocked. This effect is even more pronounced for larger σ/μ ratios. As a result, the average flow rate of carried traffic will decrease and therefore the blocking probability (in terms of flows) will decrease as well. The theoretical analysis of this issue appears to be difficult and future work will have to address this issue and the significance of the flow blocking probability in this situation. The current model is suitable for the case where links accommodate a higher number of flows and in situations where low blocking is required, for example, resource dimensioning. In all cases, the model provides better agreement than a simple average calculation using the Erlang model.

ACKNOWLEDGEMENT

The authors would like to thank Sanjay K. Bose for valuable discussions and the Australian Telecommunications Cooperative Research Centre (ATcrc) for their financial assistance of this work.

REFERENCES

1. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. *Requirements for Traffic Engineering Over MPLS.* IETF, September 1999. RFC 2702.
2. B. Fortz and M. Thorup. Optimizing OSPF/IS–IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, May 2002.
3. J. Murphy, R.J. Harris, and R. Nelson. Traffic engineering using OSPF weights and splitting ratios. *In Proceedings of Sixth International Symposium on Communications Interworking of IFIP - Interworking 2002, Fremantle WA, October 13-16*, 2002.
4. G. R. Ash. *Dynamic Routing in Telecommunication Networks.* McGraw-Hill, 1997.
5. J. Regnier, F. Bedard, J. Choquette, and A. Caron. Dynamically controlled routing in networks with non-DCR-compliant switches. *IEEE Communications Magazine*, pages 48 –52, July 1995.
6. R.J. Gibbens, F.P. Kelly, and P.B. Key. Dynamic alternative routing - modelling and behaviour. *Proc 12th International Teletraffic Congress (ITC 12), Turin Italy*, 1988.
7. K. Kawashima and A. Inoue. State- and time-dependent routing in the NTT network. *IEEE Communications Magazine*, pages 40–47, July 1995.
8. Caspian Networks. *Flow-State Routing: Rationale and Benefits*, July 2004. White Paper - www.caspiannetworks.com/files/Apeiro_Flow_State.pdf.
9. D. Thaler and C. Hopps. *Multipath Issues in Unicast and Multicast Next-Hop Selection.* IETF, November 2000. RFC 2991.
10. A.A. Kist and R.J. Harris. Scheme for alternative packet overflow routing (SAPOR). *In IEEE Workshop on High Performance Switching and Routing (HPSR 2003), Turin, Italy*, June 2003.
11. J.S. Kaufmann. Blocking in a shared resources environment. *IEEE Transactions on Communications*, 29(10):1474–1481, 1981.
12. J.W. Roberts. A service system with heterogeneous user requirments. *In Performance of Data Communication Systems and Their Applications*, pages 423–431, 1981. North-Holland.
13. T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary poisson view of internet traffic. *INFOCOM 2004, 7-11 March*, 3:1558 – 1569, 2004.
14. R. Syski. *Introduction to Congestion Theory in Telephone System.* North-Holland, 2nd ed edition, 1986.
15. M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, January 1998.