TELETRAFFIC AND DATATRAFFIC
in a Period of Change, ITC-13
A. Jensen and V.B. Iversen (Editors)
Elsevier Science Publishers B.V. (North-Holland)
© IAC, 1991



# DETERMINATION OF LOAD-SERVICE CURVES FOR DISTRIBUTED SWITCHING SYSTEMS: PROBABILISTIC ANALYSIS OF OVERLOAD-CONTROL SCHEMES

Arthur W. Berger

AT&T Bell Laboratories
Holmdel, New Jersey, USA

For the engineering, operation and administration of switching systems, it is desirable to be able to quickly and accurately estimate the grade of service for different traffic loadings and for hypothetical scenarios and thus to be able to answer what-if questions. This paper presents a probabilistic model that meets this need for a class of overload controls in distributed switching systems. The model is modular and can capture the salient features of a variety of throttle and monitor designs. The model accurately calculates the probability a call is blocked given hypothetical traffic mixes, customer retry probabilities, load imbalances and load variations during the busy hour.

## 1. INTRODUCTION

For the engineering, operation and administration of switching systems, it is important to know the grade of service attained for different traffic loadings. This information is provided by load-service curves that plot a performance measure, such as the probability a call is blocked or dial tone delay, versus the offered load. Load-service curves are used for both the component parts of switching systems, as well as the overall switching system and telecommunication networks. They are cited extensively in the literature, and examples most relevant to the present paper include Briccoli et al., who used load-service curves in describing the performance of a distributed switching system, [1]. Basu et al. and Tran-Gia used them to show the performance of overload control designs for switching systems, [2,3]. Forys et al. examined the "efficacy of using artificially generated 'load box' traffic to determine load-service relationships" for digital switching systems, [4].

Major factors that determine the load-service curves for digital switching systems are: (1) the real-time capacity of the system, which in turn depends on the mix of different traffic types, and (2) the performance of overload controls that regulate the admission of new calls when the offered load is beyond system capacity. As pointed out by Kappel and Stone, [5], it is both difficult and crucial to quantify the performance of a system's overload control plan. Direct evidence from measurements of a switching system in overload is most useful. The measurements can be obtained in a laboratory setting from system tests and in the field from switching systems in operation. Simulation studies complement the direct measurements by providing evidence of performance under hypothetical scenarios, such as load levels that can not be attained in a laboratory or not yet experienced by switching systems in the field. Analytic models complement both the direct measurements and the simulation studies. After validation with measurements or simulation, the analytic model becomes a valuable tool for answering "what-if" questions. Load-service curves can be generated much more quickly with an analytic model than with simulation. The probabilistic model described herein easily calculates families of load-service curves that show the impact of different traffic mixes, customer retry probabilities, load imbalances and load variations during the busy hour.

## 2. THE CLASS OF OVERLOAD CONTROLS

The class of overload controls is for a star topology, distributed switching system where the limiting resource for call processing is the central module (CM), see Figure 1.

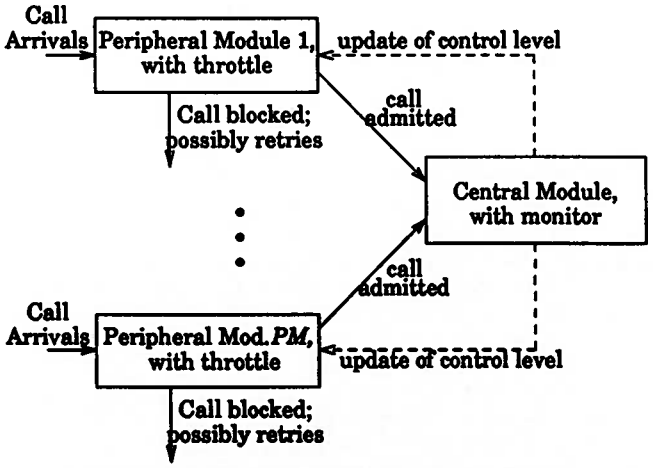


Figure 1. Diagram of overload-control scheme in a distributed switching system.

The monitor to detect the overload is located at the CM and the throttles that regulate the arriving call requests are located at the peripheral modules (PMs). The main processor at the CM is responsible for a portion of the call setup process, as well as Operations, Administration and Maintenance (OA&M) and other tasks. Although call processing has higher priority than OA&M, the amount of real-time devoted to call

processing is restricted in order that other functions obtain at least a given amount of processor time. After monitoring the workload for call processing that arrives over a set interval of time, the CM signals to all of the peripheral modules whether and to what extent call requests are to be throttled during the next interval. The signal (the control level) is a non-negative integer, where the higher the value, the more severe the overload. This structure allows for some separation in the design of the monitor and the throttle. It also enables the methodology presented in the next section to apply to a variety of monitor and throttle designs.

For ease of exposition, we consider a particular scheme for the monitor and for the throttles, and then in Sections 3.2.1 and 3.3.1 we describe a range of alternatives. The particular design for the monitor uses two thresholds to provide hysteresis. The monitor turns on, or increments to a higher level, if the workload arriving during a monitoring interval exceeds a given threshold. Likewise, it decrements one level, or turns off, if the workload is below a second, and lower, threshold.

The throttle is a rate-control throttle based on token banks. Token banks are counters that increment up periodically and decrement at call arrivals. In particular, internal to the throttle, tokens arrive evenly spaced from an infinite source. The rate that tokens arrive is determined by the current control level from the monitor at the CM. The token bank has finite capacity, and tokens that arrive to a full bank are lost. If the token bank contains a token when a call arrives, then the call is allowed to pass through the throttle and the bank is decremented by one token. If the bank does not contain a token when a call arrives, then the call is blocked and rejected.

## 3. DESCRIPTION OF THE ANALYTIC MODEL

We assume that calls arrive as a Poisson process, and we make the following approximations to obtain a reasonably accurate, though simple, analytic model of the overload control schemes.

(A1) Let the steady state departure rate from a throttle, given a constant control level, approximate the true departure rate within monitoring intervals with the same control level.

(A2) Let the distribution for the number of calls admitted during a monitoring interval depend on the control level and on the Poisson call arrival process.

(A3) Let the distribution in (A2) be Poisson.

For rate-control throttles, approximation (A2) ignores the dependence on the state of the token banks at the start of the monitoring interval. The Poisson assumption in (A3) is not critical to the methodology. For parameter values of interest, the expected number of calls admitted during a monitoring interval is of the order of a thousand. For these parameters, the Poisson distribution is close to the normal distribution. (If the normal distribution were assumed with the same mean and variance as for the Poisson, then the predicted blockings would be unchanged to 3 significant figures.) Rather, the important implication of approximation (A3) is that the variance-to-mean ratio for the number of calls admitted equals one. This is inaccurate for rate-control throttles, as the ratio is less than one.

Despite the errors introduced by approximations (A1) - (A3), the computed blocking probability is typically within a percentage point of that from a discrete-event simulation, Section 4.1. This good agreement can be due to a partial cancellation of the errors introduced by the approximations: the steady-state assumption (A1) leads to a more "regular" arrival process to the CM, while the assumed higher than actual variance-to-mean ratio in (A3) leads to a less regular arrival process. Although (A3) is used, in ongoing work, we wish to better characterize the arrival process to the CM, given rate-control throttles. One could start with the departure process from a single PM (which is equivalent to the departure process from a D/M/1/K queue, where the entities queued are the tokens, [6,7]) and then approximate the superposition.

### 3.1 Calculation of the Blocking Probability

Let $\lambda_i$ equal arrival rate of calls to peripheral module $i$, $i \in \{1, ..., PM\}$ and let the control levels be numbered $\{0, 1, ..., N\}$. Control level 0 denotes that the throttle is off: all calls are admitted to the CM and the throttles are inactive. For control levels 1 through $N$, the throttle is on, and level 1 corresponds to the most relaxed throttle setting (highest arrival rate of tokens), and control level $N$ corresponds to the strictest throttle setting (lowest arrival rate of tokens). For a given vector of arrival rates $\lambda = (\lambda_1,...,\lambda_{PM})$ and ignoring customer retries for the moment, the fraction of calls blocked is calculated via the following steps:

1. For each control level $j$, compute the steady state throughput and blocking at PM $i$, denoted $\lambda_i'(j)$ and $b_i(j)$ respectively, $i$=1, ..., $PM$, $j$=0, ..., $N$. See Section 3.2 for details. From assumption (A1), these steady state throughputs and blockings are used to approximate the throughputs and blockings during a given monitoring interval. Note that $\lambda_i = \lambda_i'(0) > \lambda_i'(1) > ... > \lambda_i'(N)$, and $0 = b_i(0) < b_i(i) < ... < b_i(N)$. The sum of the throughputs from the PMs is the arrival rate to the CM: $\lambda'(j) \equiv \sum_{i=1}^{PM} \lambda_i'(j)$.
2. Given the arrival rate to the CM conditioned on the control level, $\lambda'(j)$, and using assumptions (A2) and (A3), estimate the fraction of time the throttle is at control level $j$, denoted $\alpha(j)$. See Section 3.3 for details.
3. The blocking at PM $i$, denoted $b_i$, is easily obtained from the conditional blocking given the control level, $b_i(j)$, of step 1 and the fraction of time the throttle is at a given level, $\alpha(j)$, of step 2:

$$b_i = \sum_{j=1}^{N} b_i(j) \cdot \alpha(j).$$

4. Given the $b_i$'s, the overall blocking for the office is equal to $\sum_{i=1}^{PM} \lambda_i b_i \; / \; \sum_{i=1}^{PM} \lambda_i$.

The model incorporates customer reattempts via a simple retry model: each time a customer is blocked, he/she retries with a given probability. The resulting total offered load (first attempts plus retries) is approximated with a Poisson process with an arrival rate equal to the first offered load plus the portion of total offered load that is blocked and retries. (For a more detailed retry model, see Reeser, [8].) As the blocking probability is, itself, a function of the total arrival rate, a simple iteration is used. Let *retry* denote the probability a blocked customer retries. Let $\lambda_i$ equal total arrival rate to PM $i$ (first offered plus retries), and let $\hat{\lambda}$ equal the vector of the $\hat{\lambda}_i$'s. Indicate the dependence of $b_i$ on $\hat{\lambda}$ via $b_i(\hat{\lambda})$. The $\hat{\lambda}_i$'s are given implicitly by: $\hat{\lambda}_i = \lambda_i \; / \; [1 - b_i(\hat{\lambda}) \cdot retry]$, $i = 1, \ldots, PM$. To compute $\hat{\lambda}$ one can use the iteration: $\hat{\lambda}_i(k+1) = \lambda_i \; / \; [1 - b_i(\hat{\lambda}(k)) \cdot retry]$, where $\hat{\lambda}(k)$ is the $k^{th}$ iterate and where $b_i(\hat{\lambda}(k))$ is determined via steps 1-3 above.

### 3.2 Calculation of the Throughput & Blocking at each PM, for Given Control Level

Let $r(j)$ equal arrival rate of tokens at each of the PMs, given the $j^{th}$ control level, $j \in \{1, \ldots, N\}$. In steady state and for a constant control level, by definition, $\lambda_i'(j) = \lambda_i \times [1 - b_i(j)]$. Moreover, since each call that passes through the throttle requires a token and tokens either depart with a call or are lost from a full token bank, then $\lambda_i'(j)$ also equals the rate that tokens depart with calls. Thus:

$$\lambda_i'(j) = \lambda_i \times [1 - b_i(j)] \qquad (1a)$$

$$= r(j) \times [1 - \text{Prob(token is blocked)}]. \qquad (1b)$$

Using an embedded Markov chain at epochs just prior to token arrivals, and assuming calls arrive as a Poisson process, we determine the Prob(token is blocked), [6]. The throughput and blocking of calls is then known trivially from equation (1). One can get closed form expressions for the Prob(token is blocked), although the algebra becomes tedious as the token-bank capacity increases. In any case, it is easily solved for numerically. For small values of the token-bank capacity, $C$, and letting "$a$" abbreviate $\lambda_i / r(j)$:

For $C = 1$, $\quad \lambda_i'(j) = r(j)(1 - e^{-a})$

For $C = 2$, $\quad \lambda_i'(j) = r(j)\left[1 - \dfrac{e^{-2a}}{1 - a\, e^{-a}}\right]$

For $C = 3$, $\quad \lambda_i'(j) = r(j)\left[1 - \dfrac{e^{-3a}}{1 - 2a\, e^{-a} + \frac{1}{2}a^2\, e^{-2a}}\right]$

For $C = 4$, $\quad \lambda_i'(j) = r(j)\{1 - e^{-4a} \; / \; (1 - 3a\, e^{-a} + 2a^2\, e^{-2a} - \frac{1}{6}a^3\, e^{-3a})\}$

For $C = 5$, $\quad \lambda_i'(j) = r(j)\{1 - e^{-5a} \; / \; (1 - 4a\, e^{-a} + \frac{9}{2}a^2\, e^{-2a} - \frac{4}{3}a^3\, e^{-3a} + \frac{1}{24}a^4\, e^{-4a})\}$

#### 3.2.1 Other Throttle Designs

If other throttle designs were used, then $\lambda_i'(j)$ and $b_i(j)$ would be different functions of $j$. For example, with a percent-blocking throttle, each arrival is blocked with a given probability. In a typical design, $b_i(j)$ would be a predetermined parameter value for each $j$, for instance, $b_i(1)$ might be 0.95. From the definition of throughput, $\lambda_i'(j)$ is still given by equation (1a). As another example, a call-gapping throttle closes for a deterministic time interval, the gap size, $g$; after this interval, the next call to arrive passes through and the throttle again closes for the interval $g$. The gap-size could be a set parameter for each control level, say $g(j)$, and $b_i(j) = \lambda_i g(j) \; / \; [1 + \lambda_i g(j)]$, for Poisson call arrivals. A trivial, third example is the on-off (a.k.a. bang-bang) throttle, where there are two control levels and $b_i(0) = 0$ and $b_i(1) = 1$. Of course, all of the above schemes could have parameter values that depended on the PM $i$, as well as the control level $j$.

### 3.3 Calculation of Fraction of Time Throttle is at Given Control Level

Let $L_n$ equal the control level during the $n^{th}$ monitoring interval, $L_n \in \{0,1, \ldots, N\}$. When $L_n = j$ $(j > 0)$, then the token arrival rate to each bank is $r(j)$. As described in Section 2., at the end of a monitoring interval, $L_n$ either increments up one level, does not change, or decrements one level. Thus, $L_{n+1}$ is determined by $L_n$ and the workload that arrived during the $n^{th}$ monitoring interval. Using the approximation (A2), the workload that arrived during the $n^{th}$ monitoring interval just depends on $L_n$ and the Poisson call arrival process. Hence, $L_n$ is the state of a Markov chain. The equilibrium vector for this Markov chain is the fraction of time the throttle spends in each level, i.e., the vector $\alpha \equiv (\alpha(0), \ldots, \alpha(N))$. The remainder of this section presents the above points in greater detail.

Let $N_n$ equal the number of calls admitted to the CM during $n^{th}$ monitoring interval. $N_n$ depends on $L_n$ and the state of the token banks at the start of $n^{th}$ monitoring interval. From approximation (A2), we ignore the latter dependence and write $N_n$ as $N(j)$ to denote the number of calls admitted to the CM during an arbitrary monitoring interval, given that the control level is $j$. By approximation (A3), $N(j)$ has a Poisson distribution. Let $X_i$ equal the processing time of the $i^{th}$ call admitted during a monitoring interval. Assume the $X_i$'s are independent and identically distributed (i.i.d.) and are independent of $N(j)$. The distribution for $X_i$ models the traffic mix. For instance, the processing

time for a particular type of call might be roughly deterministic, in which case the distribution for $X_i$ could be chosen to be discrete with each point mass corresponding to a different type of call. The workload admitted to the CM during the $n^{th}$ monitoring interval is $\sum_{i=1}^{N(L_n)} X_i$.

Let $\rho^o$ be the occupancy allocated for call processing at the CM. For hysteresis, let $\rho^u$ be the threshold occupancy for the throttle to turn on, or increment to stricter control level. Likewise, let $\rho^d$ be the threshold occupancy for the throttle to decrement to a more relaxed control level; $\rho^d < \rho^o < \rho^u$. With $\tau$ representing the length of a monitoring interval, the monitor operates as follows:

$$L_{n+1} = \begin{cases} \min(L_n + 1, N) & \text{if } \sum_{i=1}^{N(L_n)} X_i > \tau\cdot\rho^u \\ \max(L_n - 1, 0) & \text{if } \sum_{i=1}^{N(L_n)} X_i < \tau\cdot\rho^d \\ L_n & \text{otherwise} \end{cases} \quad (2)$$

As an aside, to implement equation (2) presumes that the CM knows the value of $\sum_{i=1}^{N(L_n)} X_i$ at the end of the monitoring interval or, equivalently, that the processing times are known when the call arrives to the CM. If this does not pertain, then a close approximation would be to monitor the work *processed* during the interval.

Let $p(j)$ equal the probability the throttle decrements, given that the throttle setting has been at level $j$ during the just completed monitoring interval, $j \in \{1, ..., N\}$. Likewise, let $q(j)$ equal the probability the throttle increments given the throttle has been at level $j$, $j \in \{0, ..., N\text{-}1\}$.

$$p(j) = \text{Prob}\left(\sum_{i=1}^{N(j)} X_i > \tau\rho^u\right) \quad (3a)$$

$$q(j) = \text{Prob}\left(\sum_{i=1}^{N(j)} X_i < \tau\rho^d\right) \quad (3b)$$

Thus, to determine the $p(j)$'s and $q(j)$'s, one needs to calculate the tail probabilities of $\sum_{i=1}^{N(j)} X_i$. Since the $X_i$'s are i.i.d. and independent of $N(j)$, then the Laplace-Stieltjes Transform (LST) of $\sum_{i=1}^{N(j)} X_i$ equals the probability generating function of $N(j)$ evaluated at the LST of $X_1$. Thus, one can use a numerical algorithm such as [9] to obtain the tail probabilities from the LST, and thus calculate $p(j)$ and $q(j)$.

The illustrative example in Section 4. considers the simple scenario where the processing times are deterministic. In this case, equation (3) simplifies to:

$$p(j) = \text{Prob}(N(j) > \tau\rho^u \text{ / processing time})$$

$$q(j) = \text{Prob}(N(j) < \tau\rho^d \text{ / processing time})$$

Given the $p(j)$'s and $q(j)$'s, the equilibrium vector for the Markov chain $\{L_n\}$ can be determined by the well-known iteration:

$$\alpha(j+1) = \frac{q(j)}{p(j+1)}\,\alpha(j) \quad j = 0,...,N-1. \quad (4a)$$

$$\text{where } \alpha(0) = \left[1 + \sum_{k=1}^{N}\prod_{j=1}^{k}\frac{q(j-1)}{p(j)}\right]^{-1} \quad (4b)$$

Although the iteration (4) is standard, it is numerically awkward to use for the present model. Given any vector of arrival rates, $\lambda$, then, with high probability, the throttle moves amongst only a subset of the control levels. Thus, numerically some of the $p(j)$'s and $q(j)$'s are computed to be zero, and the Markov chain is reducible, with one irreducible class. To use a variant of (4), one would first need to determine which states are in the irreducible class. This is not necessary with the following alternative algorithm, which also correctly computes to be zero those $\alpha(j)$'s that correspond to transient states. First, compute:

$$\alpha(0) = \prod_{k=1}^{N} p(k) \quad (5a)$$

$$\alpha(j) = \prod_{k=0}^{j-1} q(k)\cdot\prod_{k=j+1}^{N} p(k) \quad \text{for } j=1,...,N-1 \quad (5b)$$

$$\alpha(N) = \prod_{k=0}^{N-1} q(k) \quad (5c)$$

Second, normalize $\alpha$ to 1.

### 3.3.1 Other Monitor Designs

When the throttle turns on, it need not start at level 1; it could enter at a higher level to yield a faster transient response. (However, the deleterious affect of false alarms would also increase.) Also, the monitor could have more than two thresholds, and the control level could increment or decrement by more than one level. With these changes, equations (4) and (5) would no longer hold. However $\{L_n\}$ would still be a Markov chain and the equilibrium vector could still be determined numerically.

## 4. ILLUSTRATIVE RESULTS

To illustrate the model, consider a generic example with the hypothetical parameter values given in Table 1 below. An office capacity of 250,000 calls/hour means that if that load were admitted to the CM, then the occupancy from call processing would be $\rho^o$. As a base case, assume no customer reattempts, a balanced loading across the PMs, and Poisson call arrivals with constant $\lambda$ throughout the busy hour. Each of these

assumptions is relaxed in turn in Sections 4.2 - 4.4.

| Parameter | Value |
|---|---|
| Office capacity | 250,000 calls/hour |
| Number of PMs | 50 |
| Token-bank capacity | 5 tokens |
| No. of control levels | 8 |
| $r(1)$ | 7,000 tokens/hour |
| $r(2)$ | 6,500 tokens/hour |
| $r(3)$ | 6,000 tokens/hour |
| $r(4)$ | 5,500 tokens/hour |
| $r(5)$ | 5,000 tokens/hour |
| $r(6)$ | 4,000 tokens/hour |
| $r(7)$ | 3,000 tokens/hour |
| $r(8)$ | 500 tokens/hour |
| $\tau$ | 20 seconds |
| $\rho^u$ | 52% |
| $\rho^o$ | 50% |
| $\rho^d$ | 48% |

Table 1.

### 4.1 Accuracy of Model

Table 2. compares the blocking probability predicted by the model with that from a discrete-event simulation. Consistent with the model, the simulation uses Poisson call arrivals, the token banks and monitor of Section 2. and the parameter values in Table 1. However, the simulation does not use the simplifying assumptions (A1) to (A3); rather, it tracks the progress of each call.

| OFFERED LOAD Arrival Rate / Office Capacity | FRACTION BLOCKED Model | Simulation 95% conf. int. | Ideal |
|---|---|---|---|
| 1.00 | .014 | .005 ± .001 | .000 |
| 1.02 | .022 | .020 ± .002 | .020 |
| 1.05 | .047 | .047 ± .002 | .048 |
| 1.10 | .091 | .092 ± .001 | .091 |
| 1.20 | .167 | .171 ± .002 | .167 |
| 1.50 | .337 | .336 ± .002 | .333 |
| 2.00 | .503 | .500 ± .001 | .500 |

Table 2.

Table 2. also contains the "ideal" blocking, which is defined to be the fraction blocked such that the admitted load equals the office capacity, given that the offered load is above capacity. I.e., the ideal blocking = max{ 0, 1 - (office capacity) / (call arrival rate) }. Note that although the call arrival rate when averaged over a period such as an hour may be below the capacity and the ideal blocking (as defined above) is zero, congestions can still occur during shorter periods of a few minutes. Depending on the circumstances, the activation of the overload control may or may not be appropriate. For the throttle design herein, the analytic model does estimate the blocking during these random congestions.

Table 2. shows that overall the blocking predicted by the model is close to the true (simulated) blocking. However, the model does give a high estimate for the blocking at capacity: 1.4% versus .5%. Similar results were obtained with other parameter values. Comparison of the predicted blocking with the ideal blocking shows that the throttle performs quite well.

### 4.2 Customer Retries

Customer retries can significantly affect the blocking seen at the switching system. Figure 2. shows the dramatic increase in blocking when customers retry with a .5 or .8 probability, as compared with no reattempts. Thus, an important feature of the present model is that it enables switching system engineers and administrators to estimate the impact of customer reattempts.

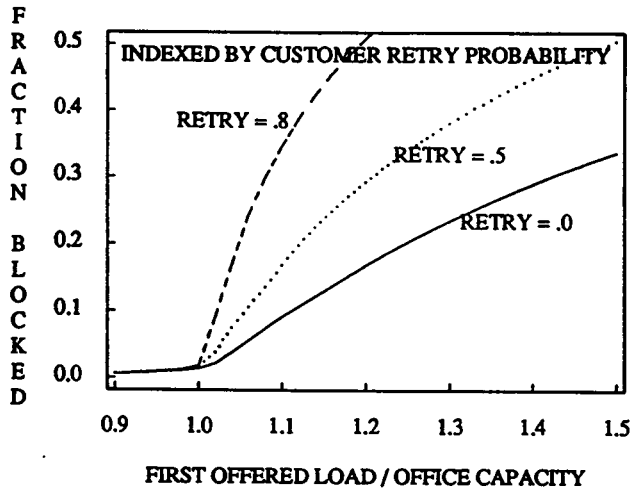


Figure 2. Load-service curves: customer reattempts.

### 4.3 Load Variation Across the Peripheral Modules

Consider a generic scenario where the PMs are partitioned into two groups, and the call arrival rate is the same to each PM within a given group but differs between the two groups. In particular, suppose one

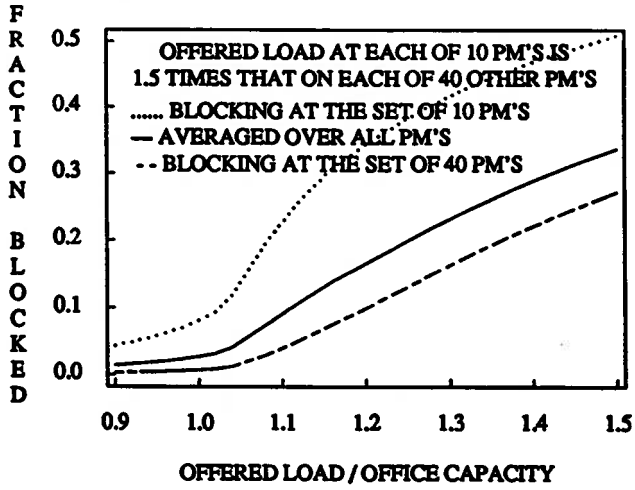


Figure 3. Load-service curves: imbalance in load.

group contains 10 PMs, the other contains 40 PMs, and the arrival rate per PM in the first group is 1.5 times the arrival rate per PM in the second group. As shown in Figure 3., the model can estimate the higher (lower) blocking on the more (less) heavily loaded PMs. As an aside, the overall blocking almost coincides with that for a balanced loading across the PMs, except the blocking is higher in the unbalanced case for loads below capacity.

### 4.4 Load Variation Within the Busy Hour

Suppose calls arrive during the busy hour as a non-stationary Poisson process. In particular, suppose that the busy hour can be partitioned into sub-periods where within each sub-period $\lambda$ is constant but between sub-periods $\lambda$ jumps in value. (This supposition could be used for analyzing counts of call attempts in switching systems that collect measurements over 15 minute intervals.) Applying the approximations (A1) - (A3) to each sub-period within the busy hour, the model estimates the blocking probability within each sub-period. The overall blocking is then estimated by a weighted sum of these probabilities, where the weights are the expected number of call attempts during the sub-period divided by the expected number of attempts during the whole period. As an illustration, suppose the call arrival rate is at a high value during the first 15 minutes of the busy hour and then drops to a lower value during the remaining 45 minutes.

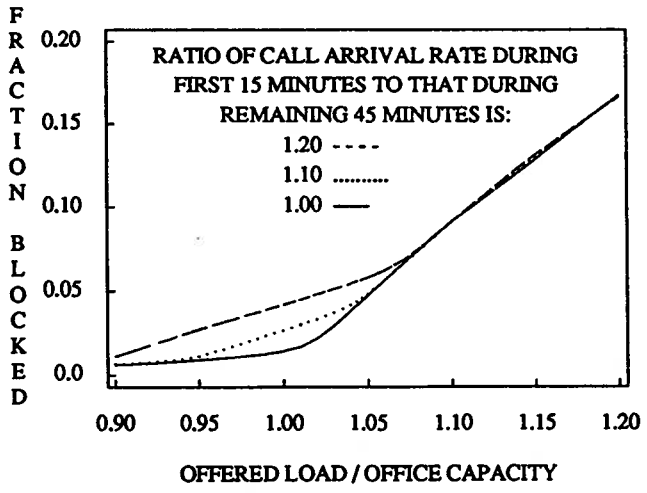


Figure 4. Load-service curves: load variation in busy hour

As shown in Figure 4., for loads around capacity the blocking is higher than for the case of constant arrival rate. Mathematically, one is computing a linear combination of the blockings from the case of constant arrival rate. (Note: the axes are scaled differently from Figures 2. and 3.)

## 5. CONCLUSIONS

A probabilistic model has been presented that calculates load-service curves for a class of dynamic overload controls in distributed switching systems. For rate-control throttles at the peripheral modules and a workload monitor with hysteresis at the central module, we showed that the blocking predicted by the model matches closely with that from a discrete-event simulation. Also, the comparison of the predicted blocking with the ideal blocking showed that the overload-control scheme performs quite well. Illustrative load-service curves were presented for hypothetical scenarios of customer retries, imbalances in load across the peripheral modules and variations in load within the busy hour.

## ACKNOWLEDGMENTS

I would like to thank Pat Wirth and Amir Sadrolhefazi for their constructive comments and Dave Lucantoni for the idea of using a Markov chain to model the operation of the monitor.

## REFERENCES

[1] Briccoli, A., G. Gallassi, G. Giacobbo Scavo & G. Miranda, "Performance Design of a Distributed Switching System," *Proc. 12th Inter. Teletraffic Congress*, Torino, Italy, 1988, Paper no. 2.1A.2.

[2] Basu, K., G. Gorman, O. Avellaneda & N. MacTavish, "A Real-Time Simulator for Performance Evaluation and Overload Control Design of an SPC System," *Proc. 10th Inter. Teletraffic Congress*, Montreal, Canada, 1983, Paper no. 5.2.9.

[3] Tran-Gia, P, "Analysis of a Load-Driven Overload Control Mechanism in Discrete-Time Domain," *Proc. 12th Inter. Teletraffic Congress*, Torino, Italy, 1988, Paper no. 4.3A.2.

[4] Forys, L. J., C. S. Im, & W. Henderson, "Analysis of Load Box Testing for Voice Switches," *Proc. 12th Inter. Teletraffic Congress*, Torino, Italy, 1988, Paper no. 3.3B.2.

[5] Kappel, J. G. & R. C. Stone, "Digital Switching Systems Traffic Analysis," *Proc. 12th Inter. Teletraffic Congress*, Torino, Italy, 1988, Paper no. 2.1A.1.

[6] Berger, A. W., "Overload Control Using a Rate Control Throttle: Selecting Token Bank Capacity for Robustness to Arrival Rates," *IEEE Trans. on Automatic Control*, Vol 36, 1991, pp. 216-219.

[7] Laslett, G. M., "Characterizing the Finite Capacity GI/M/1 Queue with Renewal Output," *Management Science*, Vol. 22, 1975, pp. 106-110.

[8] Reeser, P. K., "Simple Approximation for Blocking Seen by Peaked Traffic with Delayed, Correlated Reattempts," *Proc. 12th Inter. Teletraffic Congress*, Torino, Italy, 1988, Paper no. 3.1B.5.

[9] Platzman, L. K., J. C. Ammons & J. J. Bartholdi III, "A Simple and Efficient Algorithm to Compute Tail Probabilities From Transforms," *Operations Research*, Vol. 36, 1988, pp 137-144.