Overload Control Using Rate Control Throttle: Selecting Token Bank Capacity for Robustness to Arrival Rates

Arthur W. Berger

Abstract—**This note provides new insights that ease the design of rate control throttles, which are used for overload control of computer and communication networks. A key result is that if the token bank capacity is 10 or more and if jobs arrive as a Poisson process, then for many practical applications in digital switching systems and telecommunication networks, the control settings can be set independent of the arrival rate and need only be adjusted for changes in the desired departure rate.**

I. Introduction

The purpose of this note is to assist future implementations of rate control throttles by giving new insights into the role played by the capacity of the token bank.

A rate control throttle is used for input regulation in overload control of computer and communications systems to block and reject arriving jobs when the arrival rate is beyond system capacity [1], [2]. As defined herein, a rate control throttle contains a token bank where, internal to the throttle, tokens arrive at a deterministic rate from an infinite source (see Fig. 1.). This rate is the control variable of the throttle. The token bank has finite capacity and tokens that arrive to a full bank are blocked and lost. The capacity of the token bank is a design parameter that is typically constant during the operation of the throttle. If the bank contains a token when a job arrives to the throttle, then the job is allowed to pass through, and the bank is decremented by 1 token. If the bank does not contain a token when a job arrives, then the job is blocked and rejected. (Note that a rate control throttle differs from a sliding window flow control in that there is no constraint on the number of outstanding tokens; rather, tokens are used once and do not circulate back to the token bank.) Typically, during normal nonoverload conditions, the throttle is not turned on, and arriving jobs are not affected. When a monitor detects an overload, the throttle is activated and remains on until the monitor determines the abeyance of the overload. The initial responsiveness of the throttle can be tuned by the number of tokens placed in the token bank when the throttle is activated.

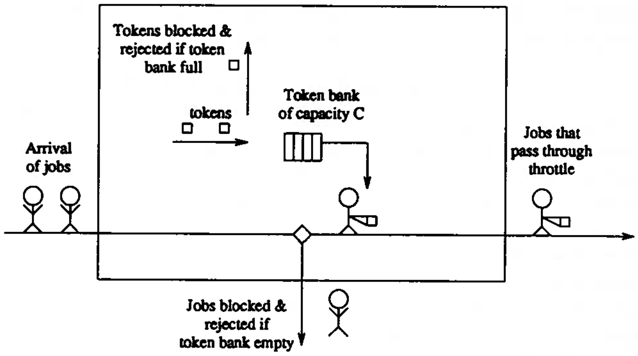
Tokens blocked & rejected if token bank full
tokens
Token bank of capacity C
Arrival of jobs
Jobs that pass through throttle
Jobs blocked & rejected if token bank empty

Fig. 1. Diagram of the rate control throttle.

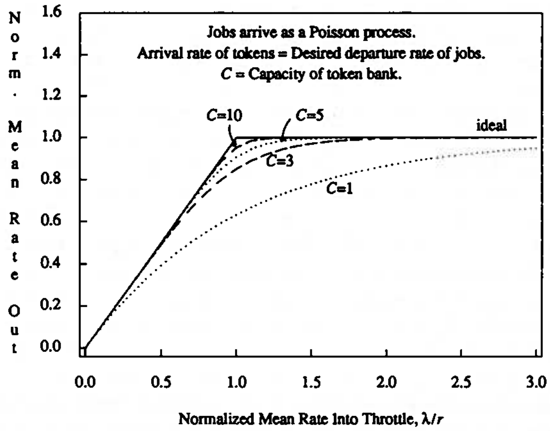

Fig. 2. Normalized mean departure rate of jobs, λ'/r versus arrival rate, given the control setting is fixed.

Doshi and Heffes study the rate control throttle in the context of a star topology network with a monitor at the central node and throttles at the peripheral nodes [1]. (They also make a comparison with sliding window flow control.) For a similar network configuration, Kumar describes a monitor that uses stochastic approximation to update the control settings [2]. In [1] and [2], the arriving jobs constitute requests to initiate a user's call or session. In contrast, Eckberg *et al.* [3] use a leaky bucket (which is almost isomorphic to a token bank) to regulate the packet flow during a session. The throttle acts as a throughput-burstiness filter for asynchronous transfer mode (ATM) cells of a broadband integrated services digital network (B-ISDN). In this application, a monitor, such as in [1], [2], is not used, and the parameters of the throttle are determined at call setup and remain fixed for the duration of the call. Cells that arrive to an empty token bank are not blocked but rather are marked, are allowed through and may be discarded if a subsequent node is congested. In [4], Sidi *et al.* also use a token bank for B-ISDN where the cells that arrive to an empty bank are not marked or blocked but rather are delayed in a job buffer. For Poisson job arrivals, they determine the Laplace–Stieltjes transform of the distribution of the cell waiting times and interdeparture times. We also have examined the case of delaying jobs in a buffer [5]. A key result is that, for jobs arriving according to a Markovian arrival process[1] and for tokens arriving according to a renewal process that is independent of the job arrival process, then the probability that a job is blocked depends on the capacity of the job buffer and the capacity of the token bank only via the sum of the two capacities.

The contribution of the present note is to show a desirable robustness feature of the rate control throttle that is of use in the design of the overall control scheme. The robustness of interest is with respect to the job arrival rate. An ideal robustness to job arrival rates would be a single control setting (token arrival rate) where the departure rate of jobs equals the desired maximum departure rate for all arrival rates above the maximum desired departure rate, and where the departure rate of jobs equals arrival rate for all arrival rates below the maximum desired departure rate (see Fig. 2.). The more robust the throttle, the less the control variable need be changed to compensate for changes in the exogenous arrival rate, including the effect of retries by customers that previously had been blocked and rejected.

In many applications, the control variable is updated periodically and the arrival rate of jobs may change markedly within an update interval. For example, if the jobs are requests to initiate a call or session, then the users may reattempt if a previous request is blocked, and the total arrival rate seen by the throttle can increase significantly. Moreover, typically, the number of possible values for the control setting is constrained and the designer must choose these values with care. The more robust the throttle, the less the designer

Manuscript received September 10, 1989; revised February 2, 1990.
The author is with AT&T Bell Laboratories, Holmdel, NJ 07733.
IEEE Log Number 9040486.

[1] Special cases of the Markovian arrival process are phase type renewal processes and Markov-modulated Poisson processes (for details, see [6]).

0018-9286/91/0200-0216$01.00 © 1991 IEEE

need be concerned with the arrival rate of jobs, and the more the designer can tune for changes in the desired departure rate.[2]

II. Analysis of Rate Control Throttle

The token bank of the rate control throttle can be viewed as a queueing system where the arrival process is the deterministic arrivals of the tokens, and the server is the first waiting space of the token bank. A service time (the time interval a token is in the first waiting space of the token bank) is determined by the job interarrival times to the throttle. For a token that arrives to a nonempty token bank and queues in the bank (i.e., is not blocked and lost), the service time will be the interarrival time of a job. In contrast, a token that arrives to an empty token bank resides at the head of the bank until the next job arrives; thus, its service time is not a full job interarrival time. (One can view the service process as continuing to operate even when the system is empty, in analogy with the job arrival process continuing to operate independent of the state of the token bank.) If the job arrival process were a renewal process with a general interarrival time distribution, then the service time of a token arriving to an empty token bank, in general, would depend on the epoch of the last job arrival. However, if the job arrival process were a Markovian arrival process then at epochs just prior to token arrivals, the joint state of the job arrival process and the number of tokens in the bank constitutes an embedded Markov chain. For the present note, we can bring out the conceptual points with a simple process: assume that jobs arrive to the throttle as a batch Poisson process with geometrically distributed batch sizes. That is, jobs arrive in batches; the size of the batch has a geometric distribution over the positive integers, and the interarrival times of batches are exponentially distributed. Due to the memoryless property of the exponential distribution, the time from the arrival of a token until the arrival of the next batch of jobs is also exponentially distributed and with the same parameter as the interarrival times of batches. If an arriving batch of jobs is larger than the number of tokens queued in the bank, then the token in service departs, and the remaining tokens are served instantaneously, and the residual number of jobs are blocked and rejected by the throttle. Thus, the token bank is equivalent to a queueing system with deterministic interarrival times, batch Poisson service times with geometric batch sizes, one server, and a finite capacity (for a detailed discussion of such queueing systems, see [7]).

Define the notation:

λ = The mean arrival rate of jobs to the throttle.
c^2 = The squared coefficient of variation of interarrival times of jobs.
λ_b = The arrival rate of batches of jobs.
q_j = The probability of j jobs in a batch.
p = The probability the interarrival time of jobs is zero.
r = The deterministic arrival rate of tokens to the token bank.
C = The capacity of the token bank.
λ' = The mean departure rate of jobs that pass through the throttle, i.e., jobs that are not blocked and rejected by the throttle.

Since the jobs arrive as a batch Poisson process with geometric batch sizes, then

$$q_j = (1-p)p^{j-1} \qquad j = 1, 2, \cdots$$

and the job interarrival time density is $p \cdot \delta(0) + (1-p)\lambda_b e^{-\lambda_b t}$, where $\delta(0)$ is the delta function. Given λ and c^2, then λ_b and p are uniquely determined and vice versa. In particular

$$\lambda_b = \frac{2\lambda}{c^2+1}, \quad p = \frac{c^2-1}{c^2+1}. \tag{1a}$$

$$\lambda = \frac{\lambda_b}{1-p}, \quad c^2 = \frac{1+p}{1-p}. \tag{1b}$$

Thus, the modeled job arrival process can capture the first two moments of hypothetical job arrival processes that are assumed to be renewal and with $c^2 \geq 1$. Moreover, speaking heuristically, over the class of renewal processes with given first two moments, batch arrivals would cause more "stress" on the system (for a discussion of this point, see [5]). Thus, the affect of $c^2 > 1$ would be more pronounced. Note that when $p = 0$, then the geometric distribution becomes degenerate, and the batch Poisson process becomes the Poisson process.

Since each job passing through the throttle requires a token and since some tokens may be blocked and lost at a full token bank, then $\lambda' \leq r$. Moreover, in steady state and for finite token bank capacities, λ' equals r times the fraction of tokens not blocked. Thus,[3]

$$\begin{aligned} \lambda' &= \lambda \times [1 - \text{Prob(job is blocked)}] \\ &= r \times [1 - \text{Prob(token is blocked)}]. \end{aligned} \tag{2}$$

At epochs just prior to an arrival of a token, the number of tokens in the token bank constitutes an embedded Markov chain on the state space $\{0, 1, 2, \cdots, C\}$. From the Markov chain, we can calculate the equilibrium probability that an arriving token sees a full bank, i.e., the probability a token is blocked. Then, using (2), we obtain the throughput λ' and the probability a job is blocked. Let:

X_k = The number of tokens in the token bank just prior to the arrival of the kth token.
N_k = The number of jobs to arrive in the interval between the kth and $k+1$st token arrivals.
A_n = Prob(n jobs arrive during 1 intertoken arrival time) = Prob($N_k = n$).
B_n = Prob(at least n jobs arrive during 1 intertoken arrival time) = Prob($N_k \geq n$).

The state evolution equations are

$$X_{k+1} = \max(0, X_k + 1 - N_k) \qquad \text{if } X_k < C.$$
$$X_{k+1} = \max(0, X_k - N_k) \qquad \text{if } X_k = C.$$

Since the intertoken arrival time is deterministic of length $1/r$, then A_n can be determined iteratively by:

$$A_o = e^{-\lambda_b/r} \tag{3a}$$

$$A_{n+1} = \frac{\lambda_b/r}{n+1} \cdot \sum_{j=0}^{n} (n-j+1) q_{n-j+1} A_j \qquad n = 0, 1, 2, \cdots \tag{3b}$$

and B_n is given by $B_n = \sum_{k=n}^{\infty} A_k$ [7]. The one-step transition probability matrix of the embedded Markov chain, denoted $\boldsymbol{P}$, is as follows:

$$\begin{array}{c|ccccccc|} & 0 & 1 & 2 & \cdots & i+1 & \cdots & C \\ \hline 0 & B_1 & A_0 & 0 & \cdots & 0 & \cdots & 0 \\ 1 & B_2 & A_1 & A_0 & \cdots & 0 & \cdots & \\ \vdots & & & & & \vdots & & \\ i & B_{i+1} & A_i & A_{i-1} & \cdots & A_0 & & 0 \\ \vdots & & & & & \vdots & & \\ C-1 & B_C & A_{C-1} & A_{C-2} & \cdots & A_{C-i-1} & \cdots & A_0 \\ C & B_C & A_{C-1} & A_{C-2} & \cdots & A_{C-i-1} & \cdots & A_0 \\ \hline \end{array} .$$

From the form of $\boldsymbol{P}$, we see that the embedded chain is irre-

[2] In a star topology network with a bottleneck at the central node and throttles at the peripheral nodes, the desired departure rate depends strongly on: 1) the number of active sources at the periphery; 2) the availability of resources at the central node, and 3) the service times of jobs.

[3] In steady state and for infinite token bank capacity, (2) continues to hold for $\lambda > r$, but does not hold for $\lambda < r$. In the latter case, tokens are entering the token bank faster than they are departing, and in "steady state" the token queue is infinite, the Prob(token is blocked) = Prob(job is blocked) = 0, and λ' equals λ and does not equal r.

ducible, positive recurrent, and aperiodic. Hence, the chain is ergodic, and its limiting distribution equals its stationary distribution. Let $\pi = (\pi_0, \pi_1, \cdots, \pi_c)$ denote the stationary distribution determined by

$$\pi P = \pi, \quad \sum_{i=0}^{C} \pi_i = 1. \tag{4}$$

Thus π_c equals the probability an arriving token is blocked, and the throughput of jobs, λ', equals $r(1 - \pi_c)$. Note that since A_n depends on λ_b and r only via the ratio λ_b/r and since $\lambda_b = \lambda(1 - p)$ [see (3) and (1b)], then the elements of P and hence π_c depend on λ and r only via the ratio λ/r.

Given the form of P, (4) is easily solved numerically. For instance, one can arbitrarily set $\pi_c = 1$, then solve for π_{c-1} using the last column of P, then solve for π_{c-2} using the second to last column, and so on, and lastly, normalize the π_i's to 1. Also, one can explicitly solve (4) to get closed form expressions for π_c and hence, λ', but the algebra becomes tedious as C increases. For $C \le 4$, we have the following:

$$\lambda' = r\left(1 - e^{-\lambda_b/r}\right), \qquad \text{for } C = 1$$

$$\lambda' = r\left(1 - \frac{e^{-2\lambda_b/r}}{1 - \frac{\lambda_b}{r}(1 - p)e^{-\lambda_b/r}}\right), \qquad \text{for } C = 2$$

$$\lambda' = r\left(1 - e^{-3\lambda_b/r}\Big/\left\{1 - 2\frac{\lambda_b}{r}(1 - p)e^{\frac{-\lambda_b}{r}} + \frac{\lambda_b}{r}(1 - p)\left[\frac{1}{2}\frac{\lambda_b}{r}(1 - p) - p\right]e^{-2\lambda_b/r}\right\}\right),$$

for $C = 3$

$$\lambda' = r\left(1 - e^{-4\lambda_b/r}\Big/\left\{1 - 3\frac{\lambda_b}{r}(1 - p)e^{-\lambda_b/r} + 2\frac{\lambda_b}{r}(1 - p) \cdot \left[\frac{\lambda_b}{r}(1 - p) - p\right]e^{-2\lambda_b/r} + \frac{\lambda_b}{r}(1 - p) \cdot \left[\frac{\lambda_b}{r}(1 - p)p - p^2 - \frac{1}{6}\left(\frac{\lambda_b}{r}\right)^2(1 - p)^2\right]e^{-3\lambda_b/r}\right\}\right).$$

for $C = 4$

III. Robustness to Arrival Rates

Suppose the control r is fixed and set equal to the maximum desired departure rate, and suppose that λ varies. For Poisson arrival of jobs, Fig. 2 shows the resulting departure rates, indexed by the token bank capacity. Since π_c depends on λ and r only via the ratio λ/r, we can define a normalized mean arrival rate λ/r and a normalized mean departure rate λ'/r and the curves in Fig. 2 depend on λ and r only via their ratio. Fig. 2 shows that by increasing the token bank capacity C from 1 to just 3, the departure rates are significantly closer to the ideal. For $C \ge 10$, the robustness is very good. Note that in all cases the deviation from ideal is greatest at $\lambda/r = 1$. When $C = 10$, this deviation is only 5%. For $C = 20$ and 30, the deviation decreases to 2.5% and 1.6%, respectively.

There is practical significance to the almost ideal robustness to variations in λ, given Poisson jobs arrivals and $C \ge 10$. In many applications in digital switching systems and telecommunication networks, the control setting r cannot be set exactly equal to the desired value because of the granularity in the possible values.

TABLE I

Percent Deviation from Ideal of Departure Rate of Jobs

Jobs Arrive as a Batch Poisson Process, Geometric Batch Sizes
Arrival Rate of Tokens = Desired Departure Rate of Jobs

Capacity of Token Bank	$\lambda/r = 1.1$, $c^2 =$			$\lambda/r = 1.5$, $c^2 =$			$\lambda/r = 3.0$, $c^2 =$		
	1	2	4	1	2	4	1	2	4
1	33%	48%	64%	22%	37%	55%	5%	14%	30%
3	11%	21%	36%	3%	10%	24%	0%	2%	11%
5	6%	13%	24%	0%	3%	12%	0%	0%	1%
7	3%	8%	18%	0%	1%	7%	0%	0%	0%
10	2%	5%	12%	0%	0%	3%	0%	0%	0%
20	0%	1%	5%	0%	0%	0%	0%	0%	0%
30	0%	0%	3%	0%	0%	0%	0%	0%	0%

Moreover, typically, the monitor for the overload only can estimate the desired maximum departure rate. Thus, as a practical matter, the above robustness of the throttle allows the monitor to update the control setting based on the estimated desired departure rate and independent of the arrival rate.

For jobs that arrive according to a batch Poisson process with geometric batch sizes, Table I shows the percent the realized departure rate is less then the ideal, for selected values of C, c^2, and λ/r. (For $c^2 = 1$, the job arrival process is Poisson.) From Table I, we see that as c^2 increases, the robustness of the rate control throttle declines; nevertheless, it remains quite good for $C \ge 10$ and $c^2 \le 2$.

Although the robustness declines as c^2 increases, the resulting lowered departure rate from the throttle may actually be an advantage. The departure process from the throttle partially retains the burstiness of the arrival process, and typically, a bursty departure process is more stressful for the downstream system, in which case, the appropriate mean departure rate from the throttle ought to be lower to compensate for the burstiness. Of course, the lowered departure rate may not be at the optimum level for the downstream system, but at least the deviation is in the direction that causes a desirable compensating effect.

A. Comparison to Percent Blocking and Call Gapping Throttles

A percent blocking throttle (also known as a proportional control throttle) blocks and rejects an arriving job with a given probability b. Thus, the mean departure rate from a percent blocking throttle is $\lambda(1 - b)$. A call gapping throttle closes for a deterministic time interval, the gap size g; after this interval, the next job to arrive passes through, and the throttle again closes for the deterministic time interval. For Poisson arrivals of jobs, the interdeparture times from a call gapping throttle have a delayed exponential distribution, and the mean departure rate is $\lambda/1 + \lambda_g$.

To compare the robustness of the throttle schemes, suppose the control variables (r, b, g) are set optimally for an arrival rate twice the desired departure rate. Suppose the controls are then held fixed and the arrival rate varies. For Poisson job arrivals, Fig. 3 plots the resulting departure rates. Note that the rate control throttle, even with a token bank capacity of 1, is more robust to variations in λ than is either percent blocking or call gapping.

IV. Conclusions

This note has examined the robustness of the rate control throttle to changes in the arrival rate of jobs λ. A rate control throttle with a token bank capacity of just 3 yields substantially more robustness to changes in λ than a capacity of 1. For typical throttle designs for the regulation of call setups and for Poisson job arrivals, if the token bank capacity is ≥ 10, then the control setting can be selected independent of λ and only needs to adapt to changes in the desired departure rate of jobs. If the arrival of jobs is bursty (coefficient of variation of interarrival times is greater than 1), then the robustness declines but may remain quite good, depending on parameter val-

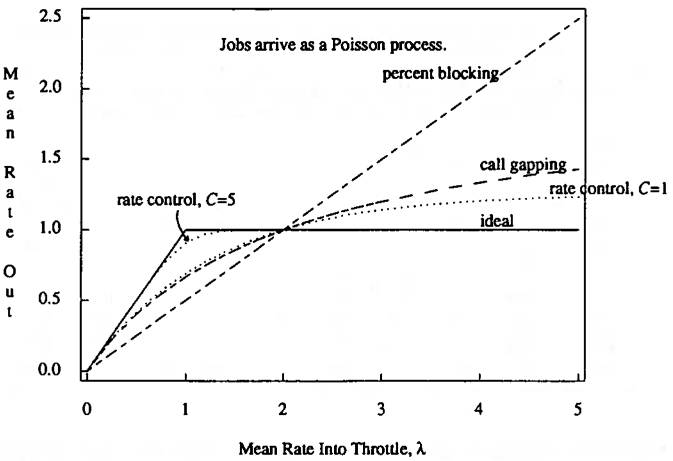

Fig. 3. Mean departure rate of jobs, λ' versus arrival rate, given the control is fixed and optimal for an arrival rate of 2 jobs per time-unit. Parameter values: $b = 0.5$, $g = 0.5$, $r = 1.255$ for $C = 1$, $r = 1.0002$ for $C = 5$.

ues. The rate control throttle with any token bank capacity is significantly more robust to λ than either a percent blocking throttle or a call gapping throttle.

In special circumstances where the maximum desired departure rate is known and unchanging, and where the job arrival process is not bursty, then the rate control throttle can be implemented without any monitor. The robustness to λ is close enough to the ideal over all values of λ that a rate control throttle that was active at all times would not unduly restrict jobs during normal, nonoverload conditions. This is particularly useful when a monitor to detect the overload is difficult or expensive to design or build. In more general circumstances, the designer can use the robustness to λ to more finely tune the control settings for changes in the desired departure rate.

However, due to transient effects, the designer should not make the token bank capacity arbitrarily large, particularly if the throttle is active at all times, or if many throttles in coordination are controlling access to a network, each once located at a different node. For λ below r to slightly above r, tokens may appropriately accumulate in the banks over a random period, and if λ were to increase suddenly, then many jobs could pass through the throttles before they begin to be restricting. The appropriate choice for the token bank capacity obviously depends on the particulars of the application; however, a range of 5 to 20 is likely to be appropriate, at least as a starting value for the regulation of call setups.

References

[1] B. T. Doshi and H. Heffes, "Analysis of overload control schemes for a class of distributed switching machines," in *Proc. 10th Int. Teletraffic Cong.*, Montreal, Canada, 1983, Session 5.2, paper 2.

[2] A. Kumar, "Adaptive load control of the central processor in a distributed system with a star topology," *IEEE Trans. Comput.*, vol. 38, no. 11, pp. 1502-1512, Nov. 1989.

[3] A. E. Eckberg, D. T. Luan and D. M. Lucantoni, "Bandwidth management: A congestion control strategy for broadband packet networks—characterizing the throughput-burstiness filter," in *Proc. Int. Teletraffic Cong. Specialist Seminar*, Adelaide, Australia, Sept. 25-29, 1989, paper 4.4.

[4] M. Sidi, W. Z. Liu, I. Cidon, and I. Gopal, "Congestion control through input rate regulation," in *Proc. GLOBECOM '89*, Dallas, TX, Nov. 27-30, 1989, pp. 1764-1768.

[5] A. W. Berger, "Performance analysis of a rate control throttle where tokens and jobs queue," in *Proc. IEEE INFOCOM* '90, June 1990, pp. 30-38.

[6] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts, "A single server queue with server vacations and a class of non-renewal arrival process," *Advances Appl. Prob.*, vol. 22, pp. 676-705, 1990.

[7] M. L. Chaudhry and J. G. C. Templeton, *A First Course in Bulk Queues*. New York: Wiley, 1983.