Systems with Multiple Servers under Heavy-tailed Workloads

Konstantinos Psounis [a], Pablo Molinero-Fernández [b], Balaji Prabhakar [c], Fragkiskos Papadopoulos [d],

[a] *Departments of Electrical Engineering and Computer Science, University of Southern California, kpsounis@usc.edu*

[b] *Department of Electrical Engineering, Stanford University, molinero@stanfordalumni.org*

[c] *Departments of Electrical Engineering and Computer Science, Stanford University, balaji@stanford.edu*

[d] *Department of Electrical Engineering, University of Southern California, fpapadop@usc.edu*

Abstract

The heavy-tailed nature of Internet flow sizes, web pages and computer files can cause non-preemptive scheduling policies to have a large average response time. Since there are numerous communication and distributed processing systems where preempting jobs can be quite expensive, reducing response times under this constraint is a pressing issue. One proposal for tackling non-preemption is through the use of multiple servers: classify jobs according to size and assign a server to each class. Unfortunately, in most systems of interest, job sizes are unknown.

An alterative is to queue all jobs together in a central queue and assign them in a FCFS fashion to the next available server. But, this has been believed to yield large response times. In this paper, we argue that this is not the case, so long as there are enough servers. The question then is: what is the right number of servers, and is this small enough to be practical?

Despite the large amount of prior work in analyzing the behavior of a central queue system, no existing models are accurate for the case of heavy-tailed size distributions. Our main contribution is a simple yet accurate model for a central queue with multiple servers. This model accurately predicts the right number of servers, and the average and variance of the response time of the system. Hence, it can be used to improve the performance of some real systems, such as multi-server supercomputing centers and multi-channel communication systems.

Key words: heavy-tailed size distribution, $M/G/K$ queue

1 Introduction

The question of whether one fast server is better than many slow servers is quite old. In traditional queueing systems, e.g. when arrivals are Poisson and services are exponential, it is easy to see that one fast server is optimal. More specifically, in an $M/M/K$ system [1] where the processing speed of each server is $1/K$, the average response time is minimized for $K = 1$ [1]. (Note that in this paper, the $G/G/K$ notation implies that each of the K servers has speed $1/K$, not 1.)

However, it has been recently observed that in numerous real systems, for example, in computer clusters and web servers, service requirements are far from exponential, they are in fact heavy-tailed [2,3,4,5]. In such systems, when it is not possible to interrupt the service of a job, multiple-server architectures outperform single-server ones. The reason is that the probability of occurrence of very long jobs is no longer exponentially small. As a result, it is quite probable for a single server to be "blocked" by a long job, making all other jobs wait for a long time until this long job has completed service.

One way to solve this problem is to introduce preemptive schemes that interrupt the long job to service shorter ones. Actually, it is well known that a system with a single server that services first the job with the shortest remaining processing time (SRPT) is optimal with respect to the average response time [6]. But preemptive policies come at a cost, and there are cases where it is impractical to interrupt jobs. For example, in a cluster of servers that run tasks with high computational and memory requirements, it is very expensive to switch between tasks.

Another way to reduce waiting times is to use many servers. The authors in [7] investigate this idea; they show that a multi-server system which assigns the next job to the next available server, known as a central-queue system, does not perform well under a fixed, small number of servers, and suggest to assign jobs to different servers according to their size. However, rarely does one know the job size *a priori*. To address this problem, the author in [8] proposes an interesting scheme that cancels a job if its service time exceeds some threshold, and services canceled jobs from scratch, in servers dedicated for long jobs. This scheme performs well in practice, but it is not work conserving.

Because of its simplicity, the multi-server central-queue system is very appealing in practice and it is widely used in a variety of real systems. Hence, it is worth to carefully investigate its performance under heavy-tail service requirements. To this end, we first make the observation that a central-queue system has good performance so long as there are enough servers to avoid concurrent

[1] A queueing system with Poisson arrivals, exponential service times, and K servers.

blocking of all of them, that is, to avoid the situation where all of them are servicing very long jobs. The question now is: how many servers does one need to achieve good performance and is this number small enough to be practical?

Unfortunately, there are no exact formulas for the average response time of a multi-server central-queue system, even for the simplified case where arrivals are Poisson and service times are independent, a system often referred to as an $M/G/K$ queue. Further, the plethora of approximations that exist for $M/G/K$ systems, see for example [1,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27] and references within, are not accurate for heavy-tail service requirements. In particular, these approximations rely heavily on the results derived for exponential service requirements, and usually do not capture the significant reduction to the average delay caused by the increase of the number of servers under heavy-tailed traffic. We present in detail this prior body of work, and verify by simulations their inability to accurately predict the behavior of an $M/G/K$ queue when service requirements have heavy tails.

Our main contribution is a simple yet accurate model for a multi-server central-queue system. The model assumes that arrivals come as a Poisson process, and it can be generalized to hold for any renewal process. It makes no assumptions for the service requirements, and it is very accurate no matter how heavy-tailed service requirements are. Interestingly, we find that the first two moments of the jobs' size distribution suffice to capture first-order dynamics of the system, as is the case for $M/G/1$ systems. It is important to note that our primary goal is to come up with an easy to use, closed-form formula for the expected delay of multi-server systems that can be use in practice. Along this lines we make a number of choices: (i) we consider heavy-tailed distributions with finite second moments, as is the case in any real system, following the paradigm of a number of other researchers [7,8,28,29,5], (ii) we are more interested in establishing the accuracy of our formulas via simulations, rather than bounding the error of the approximation using rigorous arguments, and (iii) we do not attempt to maximize accuracy, but rather to achieve high accuracy while not losing simplicity. Quite surprisingly, despite our last choice, our model is significantly more accurate than all prior models, including the ones that are quite complex and very hard to use in practice.

The organization of the paper is as follows: Section 2 shows via simulations that the average response time of a central-queue system can be very small when many slow servers are used instead of a few fast ones. Section 3 develops our model and shows its accuracy via simulations. In the next section we present a detailed survey of the large body of work that analyzes $M/G/K$ systems, compare our model to prior models, and establish its superiority. Section 5 calculates the optimal number of servers that minimizes the average response time of the system, and Section 6 concludes the paper.

2 A Single Queue with Many Servers

We consider an $M/Heavy-tailed/K$ system, i.e., a central-queue system with Poisson arrivals, heavy-tailed identically distributed job sizes that are independent from each other and the arrivals, and K servers running at rate $1/K$ each. The total system service rate is one, and the queue operates in a first-come first-served (FCFS) manner.

In general, a heavy-tailed distribution is one for which $P(X > x) \sim x^{-\gamma}$, where $0 < \gamma < 2$. A simple and popular heavy-tailed distribution is the Pareto distribution with cumulative distribution function $F(x) = 1-(m/x)^{\gamma}$, $x \geq m > 0$. Since in practice there is always some upper bound on the size of a job, a large number of researchers, see, for example, [7,8,28,29,5], have adopted the use of a bounded Pareto distribution with a very high upper bound. Following this approach, we denote by $bPareto$ a bounded Pareto distribution with cumulative distribution function $F(x) = \frac{1-(m/x)^{\gamma}}{1-(m/M)^{\gamma}}$, where $M \geq x \geq m > 0$, $M \gg m$, and $0 < \gamma < 2$. A heavy-tailed, upper-bounded distribution has a very large but finite second moment, and when applied as an input, a tiny fraction of the largest jobs comprises a sizeable fraction of the total load.

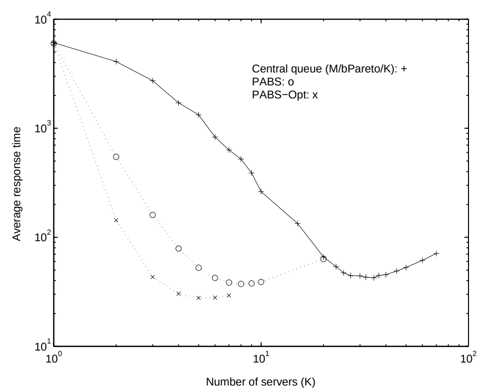

Fig. 1. Mean response time as a function of the number of servers.

Figure 1 plots the average response time for an $M/bPareto/K$ system as a function of K. (Notice that throughout the paper, the y-axis of figures plotting the average response time is normalized, i.e. it shows the average response time divided by the average job size.) The parameters of the service distribution equal $m = 1$, $M = 10^6$, and $\gamma = 1.2$. Finally, the system load, ρ, equals 0.8.

The figure also shows the performance of two schemes that assign jobs to different servers based on their size. In particular, these schemes compute $K-1$ size thresholds, and assign all jobs with size less than the smaller threshold to the first server, all jobs with size between the first and the second threshold to the second server, and so on. The first scheme, which is called pre-assigned based on size (PABS), uses the size thresholds that equalize the load among the K servers.[2] The second scheme, which is called PABS-opt, uses the size thresholds that minimize the average response time of the system.

There are two points to be observed from the plot. First, the central-queue scheme with the right number of servers performs very close to the schemes that use the size of jobs to assign them to different servers.[3] Second, as the number of available servers is increased from one, the average response time is significantly reduced for all three schemes. The reason is that the higher K is, the smaller the probability that all servers will be blocked servicing a long job. [4] It is therefore interesting to investigate how many servers a central-queue system requires to perform competitively. For this, we need a simple yet accurate model for the expected delay of an $M/G/K$ system.

Remark: We have made two choices with respect to the input: Poisson arrivals and heavy-tailed, upper-bounded independent and identically distributed sizes. These choices are in accordance to what has been observed in practice in many recent measurements of computing systems. In Web servers, it has been documented that web-page sizes are heavy-tailed [28,29,5] and that web sessions arrive as a Poisson process [30]. In Unix systems, process CPU requirements fit a heavy-tail distribution [2,3]. In the Internet, the flow-size distribution is also heavy-tailed [4]. Further, it has been measured that network sessions arrive as a Poisson process [31,32,33], and has been argued that network flows are *as if* they were Poisson [34,35]. (In particular, the equilibrium distribution of the number of flows in progress is *as if* flows arrive as a Poisson process.)

3 An Approximate Model for the Dynamics of the System

We now state the main result of this work, which we will prove later. The average response time, $E(T)$, of an $M/Heavy-tailed/K$ system can be approximated by the following expression:

[2] In [7] the authors call this scheme SITA-E and compare its performance for a fixed number of servers against the central queue scheme.
[3] We have produced Figure 1 for a wide range of γ, M, and ρ values and the results are similar. (For smaller ρ, the central-queue scheme performs even closer to PABS and PABS-opt.) Due to limitations of space we do not show these plots.
[4] However, as K increases further, the average response time deteriorates. This is so because the blocking probability becomes insignificant, and the dominant effect is then the linear decrease of the speed of the servers, which causes a linear increase in the service time.

$$E(T) \approx E(X)K + \frac{\rho}{(1-\rho)} \frac{E(X^2)}{2E(X)} \cdot \Big(1 - F_{P(\rho_l K)}(K(1-\rho_s) - 1)\Big), \qquad (1)$$

where X is the size of the jobs, $\rho = \rho_l + \rho_s = \lambda E(X)$ is the traffic intensity with ρ_l corresponding to "long" jobs and ρ_s corresponding to "short" jobs, [5] λ is the average arrival rate, and $F_{P(\lambda)}(.)$ denotes the value of the cumulative distribution function of a Poisson distribution with parameter λ.

In the rest of the section, we derive our main result and investigate how good an approximation it is. We start with the simplest of all the systems with non-negligible tails, the $M/Bimodal/K$ system. In this system, job sizes are bimodal with a probability density function $f(x) = \alpha \cdot \delta(x - A) + (1-\alpha) \cdot \delta(x - B)$, where $\delta(x) = 1$ for $x = 0$ and 0 otherwise. The size distribution is heavy-tailed when $B \gg E(X) > A$ and $\alpha \approx 1$, where $E(X)$ denotes the average job size. Later, we will extend the results to job sizes that are Pareto distributed and to job sizes that follow empirical distributions taken from real traces.

We say that the system is *blocked* when all servers are serving long jobs of size B. The system can be in two states, blocked and non-blocked. When the system is not blocked there is almost no queueing, and the response time or time in the system, T, is dominated by the service time, S, while the waiting (queueing) time, W, is insignificant. Since the service time of a job equals its size divided by the server rate, $1/K$, the average time spent in a non-blocked system equals

$$E(T|\text{non-blocked}) = E(S|\text{non-blocked}) + E(W|\text{non-blocked}) \approx E(X)K.$$

When the system is blocked, queueing can no longer be neglected, since many small jobs accumulate while the servers are occupied with long jobs. The average service time is again equal to $E(X)K$. To compute the average queueing delay we do the following approximation: We will assume that the queueing delay of a blocked system with K servers is not much different from that of a system with only one server and the same input. This is because both systems are processing work at the same rate, and when the system is blocked, no server is idle. Note that a number of prior works, e.g. [15,12,26], have made a similar approximation, in particular, they have assumed that when all servers are busy, the system can be regarded as an $M/G/1$ queue. (We regard the system as an $M/G/1$ queue when all servers are busy servicing *long* jobs.) Returning back to the derivation of the expected delay, by the Pollaczek-Khintchine formula [1] we get:

[5] Which jobs are called long and which short, is going to become precise later.

$$
\begin{aligned}
E(T|\text{blocked}) &= E(S|\text{blocked}) + E(W|\text{blocked}) \approx E(X)K + E(W|K=1) \\
&= E(X)K + \frac{\rho}{1-\rho} \cdot \frac{E(X^2)}{2E(X)},
\end{aligned}
$$

where $\rho = \lambda E(X)$ is the total system load and λ is the average arrival rate. Hence, the average time in the system is given by:

$$
\begin{aligned}
E(T) &= E(T|\text{non-blocked}) \cdot (1 - P(\text{blocked})) + E(T|\text{blocked}) \cdot P(\text{blocked}) \\
&\approx E(X)K + \frac{\rho}{1-\rho} \frac{E(X^2)}{2E(X)} P(\text{blocked}). \qquad (2)
\end{aligned}
$$

The only unknown in this expression is the blocking probability.

3.1 Blocking Probability

Let $\rho_s = \frac{\alpha A}{E(X)}\rho$ be the load caused by short jobs, and $\rho_l = \frac{(1-\alpha)B}{E(X)}\rho$ be the load caused by long jobs. In order to find the blocking probability, we first assume that ρ_s is very small. Then, we relax this assumption and study what happens when short jobs carry a non-negligible amount of work.

Blocking occurs if there are at least K arrivals of long jobs to the servers in the past BK time interval. We assume the probability of this event is close to the probability of having K arrivals of long jobs to the system in a period equal to BK. The reason is that if there is no blocking yet, the queue size is small, and any job that arrives to the system hits a server very fast. Formally, $\{\text{blocking}\} \supseteq \{\text{at least } K \text{ long arrivals to the servers in time } BK\} \supseteq \{\text{at least } K \text{ long arrivals to the system in time } BK\}$.

When short jobs carry a sizeable amount of work, they cannot be neglected as above. A simple, yet accurate way to take short jobs into account is to treat them as "background traffic". Then, because the considered time interval, BK, is a lot larger than the service time of short jobs, the work done servicing short jobs during this time interval is close to its long-term value $\rho_s \cdot BK$. The result is as if $K\rho_s$ of the servers were busy serving short jobs. Hence, the arrival of $K(1-\rho_s)$ long jobs during a time interval of BK is enough to block the system, and the blocking probability equals:

$$
\begin{aligned}
P(\text{blocked}) &\approx P(\text{at least } K(1-\rho_s) \text{ long arrivals in time } BK) \\
&= 1 - \sum_{i=0}^{K(1-\rho_s)-1} P(i \text{ long arrivals in } BK) \\
&= 1 - F_{P(\lambda(1-\alpha)BK)}(K(1-\rho_s) - 1) = 1 - F_{P(\rho_l K)}(K(1-\rho_s) - 1) \qquad (3)
\end{aligned}
$$

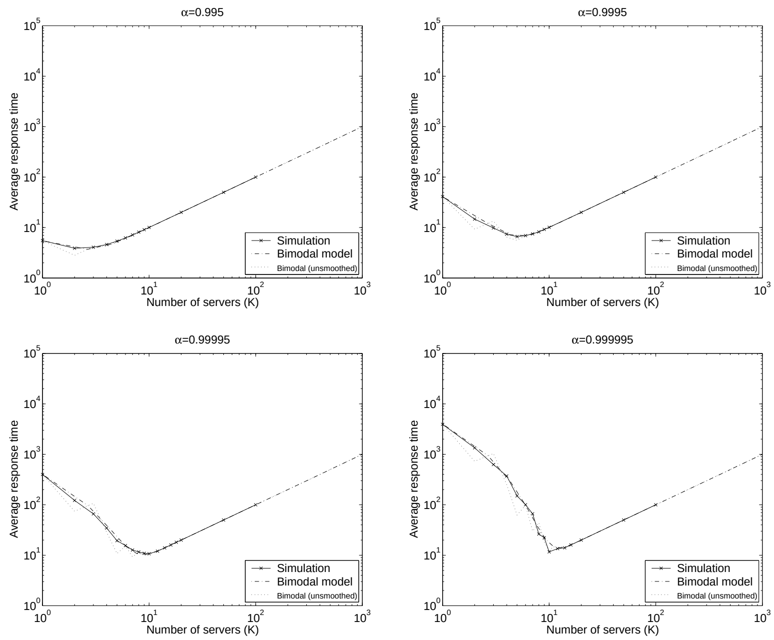

Fig. 2. Response time as obtained from simulations and the model when job sizes are distributed according to a bimodal distribution. ($\rho = 0.50$)

since the arrival process is Poisson of rate λ, and thus long jobs are also Poisson with rate $(1-\alpha)\lambda$. $F_{P(\lambda\tau)}(N)$ denotes the value of the cumulative distribution function of a Poisson distribution with parameter $\lambda\tau$, or equivalently, the probability of having at most N arrivals during a time interval τ when the arrival rate equals λ.

Combining Equations (2) and (3) we obtain Equation (1) which is our main result.

Figure 2 shows the average response time for an $M/Bimodal/K$ system with load $\rho = 0.50$, where long jobs comprise 20% of the total workload and they represent between 0.0005% and 0.5% of all jobs. The average job size equals 1500. It is evident from the plot that the model predicts the average time in the system quite accurately. Similar are the results for different loads. (Note that as α increases, the difference between A and B must also increase to keep the percentile of work carried by long jobs equal to 20%.)

Remark: $F_{P(\lambda\tau)}(N)$ is a sum between 0 and N, but the upper limit that we are using for the sum in Equation (1) is $K(1-\rho_s)-1$, which is non-integer. If we take K to be integer, every $1/(1-\rho_s)$ units we have an additional term in the sum. The result is that Equation (1) has a saw-tooth pattern that dies as K increases, as shown in the dotted line in Figure 2. If we make K take values in increments of $\Delta K = 1/(1-\rho_s)$ starting at 1, then the saw-tooth pattern is no longer present, as shown in the dash-dot line. We will use this smoothed function in all the other figures.

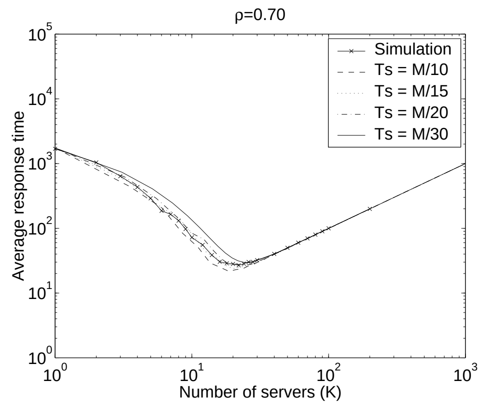

Fig. 3. Response time for various size thresholds T_s.

3.2 A More Realistic Size Distribution

As mentioned earlier, the size distribution of flows in the Internet, web pages, and process CPU requirements fits a bounded Pareto quite accurately [2,4,5]. With this in mind, in this section we extend our model to approximate $M/bPareto/K$ systems. Our goal is to compute the parameters A, B, and α of an equivalent bimodal distribution that corresponds to the bounded Pareto distribution. One can then calculate $\rho_s = \frac{\alpha A}{E(X)}\rho$ and $\rho_l = \rho - \rho_s$, and use Equation (1) to estimate the average response time as a function of the number of servers.

We choose to fit the first two moments of the two distributions for the following reasons. First, when there are many servers $E(T) = E(S) + E(W) \approx E(S) = E(X)K$, which implies that fitting the first moment suffices to have the same performance for large K. Second, when there is only one server $E(T) = E(X) + \frac{\rho}{1-\rho} \cdot \frac{E(X^2)}{2E(X)}$, which implies that fitting the second moment as well suffices to have the same performance for $K = 1$. Last, we wish to avoid using larger moments because this would increase the complexity of the procedure; larger moments are not present in Equation (1) and are extremely large in case of heavy-tailed (upper-bounded) distributions.

To fit the first two moments of the two distributions we require:

$$E(X) = \alpha \cdot A + (1-\alpha) \cdot B, \text{ and} \tag{4}$$

$$E(X^2) = \alpha \cdot A^2 + (1-\alpha) \cdot B^2. \tag{5}$$

Using the system of Equations (4) and (5) we can express A and B as a function of $E(X)$, $E(X^2)$, and α to get $A = E(X) - \sqrt{(E(X^2) - E(X)^2) \cdot \frac{1-\alpha}{\alpha}}$, and $B = E(X) + \sqrt{(E(X^2) - E(X)^2) \cdot \frac{\alpha}{1-\alpha}}$.

All that remains is to find a suitable value for α, which is the fraction of short jobs in the corresponding bimodal distribution. Intuitively, α corresponds to

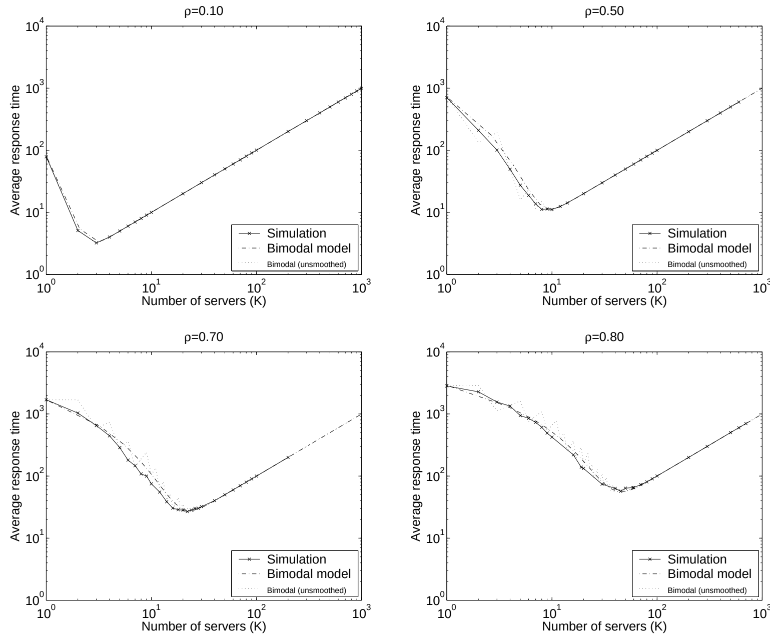

Fig. 4. Response time for different system loads as obtained from simulations and the model. Job sizes are distributed according to a bounded Pareto distribution.

the jobs that are not very large, which comprise the vast majority of all jobs. In other words, if one uses a size threshold T_s to separate short and long jobs, $\alpha = \int_m^{T_s} f(x)dx$, where $f(x)$ is the probability density function of the size distribution. By experimenting with the simulations, we found the model to be relatively insensitive to the exact value of α. This is shown in Figure 3 where the average response time in an $M/bPareto/K$ system for $\rho = 0.7$ is plotted as a function of the number of servers for various size thresholds. As a rule of thumb, the model works quite well when the size threshold dictating short and long jobs is around one order of magnitude less than the maximum job size.

Figure 4 shows the average time in an $M/bPareto/K$ system for different system loads ρ, when $m = 382.6$, $M = 10^8$, and $\gamma = 1.1$. The size threshold used equals $M/10$, that is, α is the percentile of jobs whose size is between m and $M/10$. Again, the model predicts the average time in the system quite accurately.

Remark: It is easy to see that fitting the third moment too gives one more equation, $E(X^3) = \alpha \cdot A^3 + (1-\alpha) \cdot B^3$, that can be used to compute α. If we use this equation together with (4) and (5) to map the distribution of Figure 4 to a bimodal distribution, the resulting α equals 0.999994. This is very close to the value obtained from the size-threshold approach, which equals 0.999987. (Recall that the later α-value was obtained by using $T_s = M/10$, and note that the former α value corresponds to a size threshold roughly equal to $M/5$.) Hence, due to the extra complexity associated with using the third

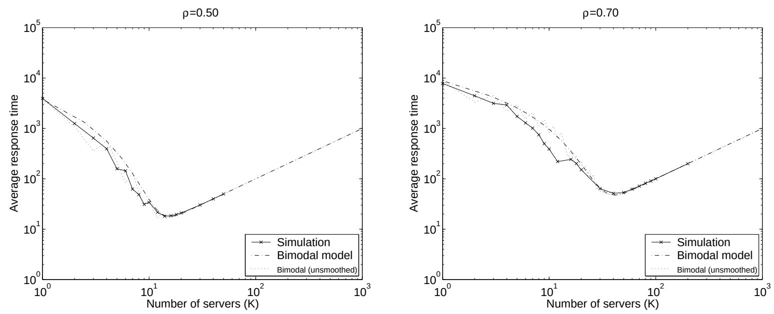

Fig. 5. Response time for different system loads as obtained from simulations and the model. Job sizes are dictated by a real traffic trace from the Internet.

moment, we do not recommend its use. The size-threshold approach is able to identify the jobs that may cause server blocking, and its accuracy is good enough.

3.3 Testing the Model Under Real Traces

In this subsection, we use job-size distributions obtained from flow-traces of real backbone links in the Internet [36] to test how good our model is in predicting the average time in the system under real traffic. We calculate from the trace the first two moments of the corresponding size distribution, compute the parameters A, B, and α of the model, and compare the average response time as obtained from the model and by running simulations using the flow-size distribution obtained from the trace. Arrivals are again Poisson. Note that in the simulation, the flow-size distribution does not fit exactly a bounded Pareto. Despite this, Figure 5 shows that the model manages to predict the average response quite accurately for a variety of system loads.

3.4 Predicting the Variance

So far, we have only studied the average response time, $E(T)$. Now, we work with its variance. First, notice that for heavy-tailed traffic the variance is very close to the second moment. Second, it is a well-known that the second moment of the queueing time in an $M/G/1$ system equals [37]:

$$E(W^2) = 2E(W)^2 + \frac{\rho}{1-\rho} \cdot \frac{E(X^3)}{3E(X)} \approx \frac{\rho}{1-\rho} \cdot \frac{E(X^3)}{3E(X)}.$$

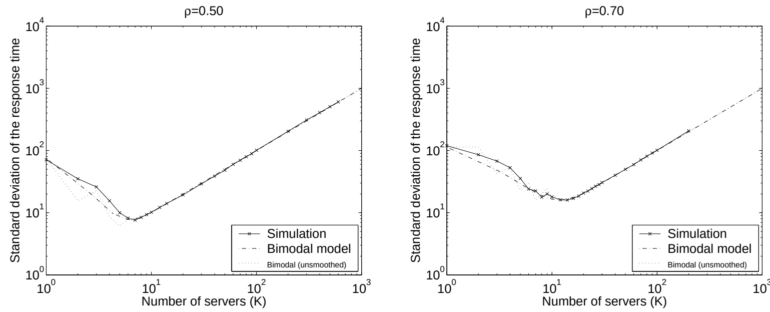

Fig. 6. Standard deviation of the response time for different system loads. The y-axis is normalized.

Using the same arguments as those used to derive Equation (2), we get:

$$E(T^2) \approx E(X^2)K^2 + \frac{\rho}{1-\rho} \cdot \frac{E(X^3)}{3E(X)} \cdot P(\text{blocking}), \tag{6}$$

where the blocking probability is calculated as before.

Figure 6 shows the standard deviation, i.e. the square root of the variance, of the response time in an $M/bPareto/K$ system for different system loads ρ, when $m = 382.6$, $M = 10^8$, and $\gamma = 1.1$. (The y-axis is normalized, that is, it plots the standard deviation of the response time over the standard deviation of the job size.) It is evident that the model can also predict the second moment of the response time. It is worth noting that the values of K that minimize the average and the standard deviation of the response time are not necessary the same, but they are not very far apart.

4 Comparison to Existing Models

In this section we compare the best approximations that exist in the literature, in terms of both accuracy and simplicity, with the one introduced in this paper. Before proceeding, recall that in our discussion the speed of each server is $1/K$ such that the total server capacity remains unchanged as K varies. Most of the results in the literature assume the speed of each server is always 1, but it is easy to change these results to account for different server speeds. We start by introducing the approximations.

The most popular approximation, which has been derived several times in the literature by various arguments, is the one obtained by Stoyan [22], Hokstad [20], Nozaki and Ross [23], Tijms et al. [26], and others, and is given by the following equation:

$$E(T) = E(X) + E(W_{M/M/K})\frac{(1+C_X^2)}{2}. \quad (7)$$

Recall that $E(W_{M/M/K})$ is the waiting time in the exponential service requirement case, for which exact closed-form formulas can be easily derived [1], and $C_X = \frac{\sigma_X}{E(X)}$ is the coefficient of variation of the service requirement.

Before presenting the rest of the approximations lets first denote by G_X the cumulative distribution function (cdf) of the service requirement, by G_e the stationary-excess cdf associated with G_X, i.e $G_e(t) = \frac{1}{E(X)}\int_0^t(1-G_X(u))du, t \geq 0$, and let $I_G(K) = \int_0^\infty(1-G_e(t))^K dt$, where $K \geq 1$ equals the number of servers.

Tijms et al. [26] attempt to improve Equation (7) by the following expression:

$$E(T) = E(X) + \left(E(W_{M/M/K})\frac{(1+C_X^2)}{2}\right)\delta, \quad (8)$$

where $\delta = 1 + (1-\rho)\left(\frac{2KE(X)}{E(X^2)}I_G(K) - 1\right)$. Observe that equations (7) and (8) differ only by the multiplicative factor δ.

Another attempt to improve Equation (7) is the following, proposed by Wang and Wolff [16]:

$$E(T) = E(X) + E(W_{M/M/K})\frac{(1+C_X^2)}{2} - \Delta, \quad (9)$$

where $\Delta = \left|P_c \cdot \left(I_G(K) - \frac{E(X^2)}{2KE(X)}\right)\right|$, and P_c is the fraction of arrivals at an $M/M/K$ queue that find K customers in the system and can be calculated recursively (see [1], p.302).

Equations (7), (8), and (9) are all $M/M/K$-based expressions, and it is easy to verify that they are exact for the $M/M/K$ case. In contrast, the following two equations interpolate between $E(W_{M/M/K})$ and $E(W_{M/D/K})$. They have been proposed by Cosmetatos [25] and Boxma et al. [24] [6], and are as follows:

$$E(T) = E(X) + C_X^2 E(W_{M/M/K}) + (1-C_X^2)E(W_{M/D/K}), \text{ and} \quad (10)$$

[6] We present the slightly modified version, suggested by Kimura [14], which accounts for the case where $K = 1$.

$$E(T) = E(X) + \frac{1 + C_X^2}{\frac{2J_G(K)}{E(W_{M/M/K})} + \frac{1-J_G(K)}{E(W_{M/D/K})}}, \quad (11)$$

where $J_G(K)$ equals 1 for $K = 1$, and it equals $\frac{K+1}{K-1}\left(\frac{(1+C_X^2)E(X)}{(K+1)I_G(K)} - 1\right)$ for $K > 1$.

The above approximations are based on the following observation. When the variance of the service requirement σ_X^2 is close to $E(X)^2$, $E(W)$ for an $M/G/K$ system is similar to the wait time in an $M/M/K$ system. When the variance is close to zero, $E(W)$ is similar to the wait time in an $M/D/K$ system. And for intermediate variance values, $E(W)$ lies between the corresponding wait time in an $M/D/K$ and an $M/M/K$ system [19].

Takahashi [38] uses the result for the $M/D/K$ system as a baseline, and accounts for the particular service requirement distribution, G_X, as follows:

$$E(T) = E(X) + \left(\frac{\mu(\alpha)}{E(X)^\alpha}\right)^{1/(\alpha-1)} E(W_{M/D/K}), \quad (12)$$

where α is such that $E(W_{M/M/K}) = \left(\frac{\mu(\alpha)}{E(X)^\alpha}\right)^{1/(\alpha-1)} E(W_{M/D/K})$, and $\mu(\alpha) = \int_0^\infty t^\alpha dG_X(t)$.

Whitt [13] considers a $GI/G/K$ system and suggests the following:

$$E(T) = E(X) + \left(\frac{C_a^2 + C_X^2}{2}\right) \Phi E(W_{M/M/K}), \quad (13)$$

where C_a is the coefficient of variation of the interarrival times (for Poisson arrivals $C_a = 1$), $\Phi = \frac{C_X^2-1}{2+2C_X^2}(1-4\gamma)e^{-2(1-\rho)/3\rho} + \frac{C_X^2+3}{2+2C_X^2}$, for $C_a^2 \leq C_x^2$, $\frac{C_a^2+C_X^2}{2} \geq 1$, and γ is the minimum of 0.24 and $(1-\rho)(K-1)\frac{\sqrt{(4+5K)}-2}{16K\rho}$. Note that both Equations (12) and (13) are exact for the $M/M/K$ case.

We finally present the simplified version of the diffusion approximation proposed by Yao [11], which is a refinement of the diffusion approximation proposed earlier by Kimura [10]:

$$E(T) = E(X) + \pi_0 \; \theta_K \; \frac{E(X)/(1-\rho)}{K(1-e^{r_K})}, \quad (14)$$

where
$\pi_0 = \left(\sum_{i=0}^{K-1} \theta_i + \theta_K/(1-\rho) + (K\rho/r_1)(e^{r_1/2} - e^{-r_1/2} - r_1)\right)^{-1}$, $\theta_i = (K\rho)^i/i!$,

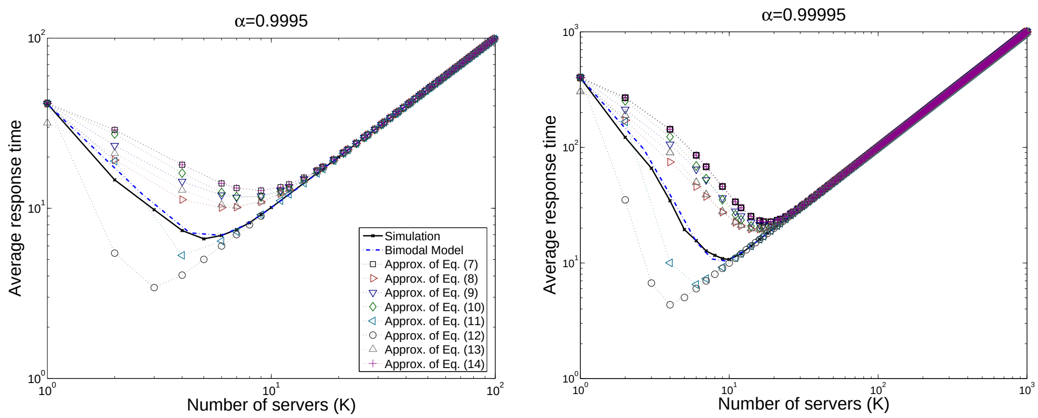

Fig. 7. Accuracy of our Bimodal model versus existing approximations. The service requirement is Bimodal. ($\rho = 0.50$, $\alpha =$ percentile of short jobs)

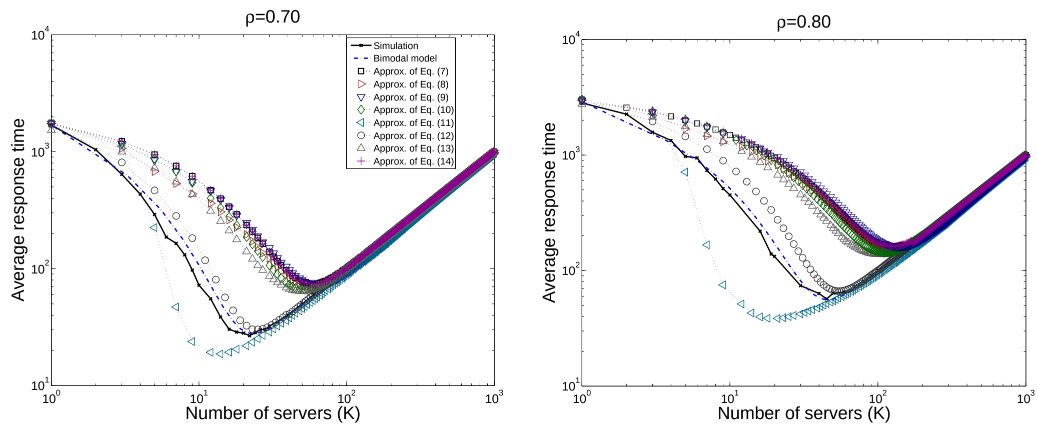

Fig. 8. Accuracy of our Bimodal model versus existing approximations. The service requirement is bounded Pareto.

$r_i = (2b_i/a_i)$, $b_i = \lambda - i\mu$, $a_i = \lambda + i\mu C_X^2$, $i = 1, ..., K$, and as usual λ is the arrival rate and $\mu^{-1} = E(X)$.

Figures 7 and 8 compare our Bimodal model versus Equations (7)–(14) for various heavy-tailed scenarios. Simulation results are also plotted for reference. Figure 7 corresponds to the scenario in Figure 2 where the service requirement is bimodal, $\rho = 0.5$, and α is the percentile of small jobs. Figure 8 corresponds to the scenario in Figure 4 where the service requirement is bounded Pareto with shape parameter equal to 1.1.

In general, as it is evident from both figures, all previous approximations are quite inaccurate. These approximations rely on the assumption that an $M/G/K$ system behaves similarly to a multiple-server system with exponential or deterministic service requirements. While this is the case when the service requirements have small variances, it is far from accurate when the service-time distribution is heavy-tailed. Note that all prior studies present numerical results for relatively small values of C_X^2, in particular for $C_X^2 \leq 9$, whereas when service times are heavy-tailed $C_X^2 \gg 1$.

Equations (8) and (9) perform a bit better than (7), especially in Figure 7. This is expected since they are improvements over (7) and take into consideration the particular service distribution. Notice that Equation (9) is not as good as (8) under high utilization, since it was derived under light traffic assumptions. Further, Equation (10) performs similarly to (7), (8) and (9).

Two approximations among prior work that give relatively good results in some cases are Equations (11) and (12). In particular, Equation (11) performs better in Figure 7 than the rest of the approximations except ours, but is bad in Figure 8, and Equation (12) performs well in Figure 8 but is quite inaccurate in Figure 7. Notice that both of these approximations incorporate into their model the particular service distribution and they are somehow hard to use in practice because of their complexity.

Equation (13) is not very accurate either. This is not surprising since the main goal in deriving this expression is to handle non-Poisson arrivals, rather than to improve over existing approximations for the Poisson case. Finally, Equation (14) performs similarly to (7). This is somehow expected since the derivation of the corresponding diffusion model uses some approximations suggested while deriving (7).

Notice that we have also compared our model to the approximations suggested by Kimura [10], Miyazawa [12] (cases 1 and 2, we left behind case 3 because it is very complicated to use), and Burman and Smith [27]. These approximations do not perform better than Equations (7)-(14) and we do not show the corresponding lines in the figures to keep them readable.

While prior approximations are inaccurate, our model is quite accurate. Considering its simplicity this is quite surprising. The key point of our approach is the observation that the system's behavior drastically depends on whether all the servers are servicing long jobs and hence they are "blocked". Depending on the intensity of long jobs and the number of servers, a system can be "blocked" for a different proportion of time, and the expected delay is affected accordingly. The parsimonious approach that we follow to compute the probability that the system is blocked yields accurate results while being easy to use in practice. Further, the simple approach that we use to map a heavy-tailed distribution into a bimodal distribution works quite well in practice. We believe this is due to the fact that only very long jobs cause server blockage, and the size threshold that we use in the mapping is enough to identify those jobs.

As a final note, given the vast differences between the accuracy of our and prior approximations, it is interesting to inspect the corresponding equations and identify where they differ. With the exception of Equations (11) and (12), the rest of the approximations appear to be clumped together in the plots, so we will compare our model to only one of them, and in particular to Equation (7) which is the most popular.

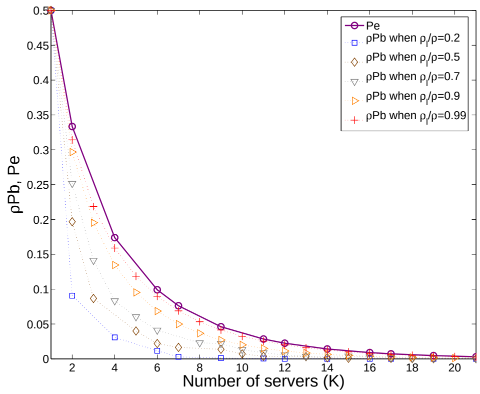

Fig. 9. Comparison of $\rho P(\text{blocked}) = \rho Pb$ and $P(\text{busy}) = Pe$. ($\rho = 0.50$)

It is well known that $E(W_{M/M/K}) = P(\text{busy})E(X)/(1-\rho)$, where $P(\text{busy})$ is the probability that all servers are busy in an $M/M/K$ queue, and can be easily computed by the stationary distribution of the queue [1]. Now, it is easy to see that Equations (1) and (7) would be the same if $\rho P(\text{blocked}) = P(\text{busy})$. Figure 9 plots $\rho P(\text{blocked})$ and $P(\text{busy})$ as a function of the number of servers for various values of ρ_l/ρ, when $\rho = 0.50$ as in Figure 7. It is evident from the plot that as the proportion of the load due to long jobs approaches one, the two terms become the same. This is an interesting result which implies that previous approximations yield similar results with our approximation only when ρ_l/ρ is close to one. (In this case prior approximations are as accurate as our model and quite close to simulation results.) When ρ_l/ρ is smaller than one, which is the case in the vast majority of real traces, see, for example, [5,36], previous approximations differ significantly from our model, and they are a lot less accurate as depicted in Figures 7 and 8. (In Figure 7 $\rho_l/\rho = 0.2$ and in Figure 8 it is between 0.1 and 0.3 depending on the size threshold used to identify long jobs.) This observation reinforces our belief that what makes our approach more accurate than prior work is the idea of "blocking" and the proper computation of the associated probability $P(\text{blocked})$.

5 On the Optimal Number of Servers

Recall that according to the model, the average time in the system is given by Equation (1). One can now differentiate this expression to find the optimal K:

$$\frac{dE(T)}{dK} = E(X) - \frac{\lambda E(X^2)}{2(1-\rho)} \cdot \frac{dF_P}{dK} = 0, \tag{15}$$

where $\frac{dF_P}{dK} = \frac{d}{dK}\sum_{i=0}^{K(1-\rho_s)-1} \frac{(\rho_l K)^i \cdot e^{-\rho_l K}}{i!}$. This derivative can be calculated using the Leibniz integral rule [39] which gives $\frac{dF_P}{dK} = (1-\rho_s) \cdot f_{P(\rho_l K)}(K(1-\rho_s) -$

$1) + \sum_{i=0}^{K(1-\rho_s)-1} f_{P(\rho_l K)}(i) \cdot (i/K - \rho_l)$, where $f_{P(\lambda)}(K)$ denotes the value of the probability mass function of a Poisson distribution with parameter λ at K. By ignoring the second term on the derivative (this term takes care of the dependence of the summation limit on K), we get $\frac{dF_P}{dK} \approx (1-\rho_s) \cdot f_{P(\rho_l K)}(K(1-\rho_s)-1)$. Hence, to compute the optimal K we need to numerically solve the equation:

$$(1-\rho_s)\frac{(\rho_l K)^{K(1-\rho_s)-1} e^{-\rho_l K}}{(K(1-\rho_s)-1)!} = \frac{2(E(X))^2(1-\rho)}{\rho E(X^2)}. \quad (16)$$

This approximation does not work well when ρ is close to one. As a result, for $\rho \geq 0.9$, one should use all the terms from the Leibniz integral rule to compute the optimal number of servers with good accuracy.

Let K_p^* be the optimal K obtained from simulating an $M/bPareto/K$ system, and K_b^* be the optimal K obtained from simulating the corresponding $M/Bimodal/K$ system described in Section 3.2. Also, let K^o be the optimal K obtained from Equation (15), and K_a^o be the optimal K obtained from using the approximation of Equation (16).

Table 5 compares these values for various system loads and size distributions. As expected from the previous plots K_b^* is very close to K_p^*. Further, for small and medium ρ, Equation (16) gives an accurate value for the optimal number of servers, while as ρ approaches one, K_a^o is not anymore a good approximation of the optimal number of servers. Even when our methods do not yield the exact optimum number of servers, the error that we incur with respect to the minimum response time is rather small. Typically, the error for K_b^* is less than 3%, and it is less than 7% in the worst case. The approximation K_a^o typically yields an error of less than 7%. However, as we mentioned before, the error is larger for high loads as shown in the table.

Table 1
Optimum number of servers for various system loads and size distributions.

ρ	m	M	γ	K_p^*	K_b^*	K^o	K_a^o	ρ	m	M	γ	K_p^*	K_b^*	K^o	K_a^o
0.5	383	10^8	1.1	10	9	9	10	0.8	383	10^8	1.1	45	45	46	59
0.5	549	10^8	1.2	7	8	7	8	0.8	549	10^8	1.2	34	32	32	38
0.5	713	10^8	1.3	6	7	6	6	0.8	713	10^8	1.3	22	21	22	25
0.7	383	10^8	1.1	22	24	22	28	0.9	383	10^8	1.1	150	146	150	217
0.7	549	10^8	1.2	18	17	17	19	0.9	549	10^8	1.2	93	93	93	128
0.7	713	10^8	1.3	12	13	13	13	0.9	713	10^8	1.3	61	58	61	77

Note that in order to compute the optimal number of servers, the only information that is needed from the traffic is the first two moments of the job-size distribution, the fraction of long jobs and the system load.

6 Conclusions

Under heavy-tailed traffic, a single fast server that operates in a FCFS manner yields very large average delays. Preemptive schemes and schemes partitioning jobs into servers based on job sizes can significantly reduce average delay. However, these schemes are often not available due to implementation constraints.

A multi-server central-queue policy that assigns the next job in FCFS order to the first available server, does not suffer from implementation constraints and has good performance if it consists of enough servers. Using simulations and analysis, we show that the required number of servers is small enough to be practical. We also provide a simple way to compute this number.

Our main contribution is the derivation of an accurate and simple to use model for an $M/G/K$ system. In contrast to prior work, our model can accurately predict the average response time of such a system when G, the jobs' size distribution, is heavy-tailed. The key point of our approach is the observation that the system's behavior drastically depends on whether all the servers are servicing long jobs and hence they are "blocked", and the accurate computation of the probability that the system is on this state.

In the derivation of the model we do a number of approximations. For example, we model the system as a single-server one when all servers are busy servicing long jobs, and we use a size threshold to map heavy-tailed distributions in corresponding bimodal distributions. These approximations make the model very simple and easy to use. Yet, our model is significantly more accurate than all previous approaches.

References

[1] R. W. Wolff, Stochastic Modelling and the Theory of Queues, Prentice Hall, 1989.

[2] M. Harchol-Balter, A. Downey, Exploiting process lifetime distributions for dynamic load balancing, ACM Transactions on Computer Systems 15 (3).

[3] W. Leland, T. Ott, Load-balancing heuristics and process behavior, in: Proceedings of the ACM SIGMETRICS Conference, 1986.

[4] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level, IEEE/ACM Transactions on Networking 5 (1) (1997) 71–86.

[5] B. Krishnamurthy, J. Rexford, Web Protocols and Practice, Addison Wesley, 2001.

[6] L. Schrage, A proof of the optimality of the shortest remaining processing time discipline, Operations Research 16 (3) (May–June 1968) 687–690.

[7] M. Harchol-Balter, M. Crovella, C. Murta, On choosing a task assignment policy for a distributed server system, Journal of Parallel and Distributed Computing 59 (2) (November 1999) 204–228.

[8] M. Halchol-Balter, Task assignment with unknown durations, Journal of the ACM 49 (2).

[9] J. Kollerstrom, Heavy traffic theory for queues with several servers, J. Appl. Prob. 11 (1974) 544–552.

[10] T. Kimura, Diffusion approximation for an M/G/m queue, Operations Research 31 (2) (1983) 304–321.

[11] D. D. Yao, Refining the diffusion approximation for the M/G/m queue, Operations Research 33 (6) (1985) 1266–1277.

[12] M. Miyazawa, Approximation of the queue-length distribution of an M/GI/s queue by the basic equations, J. Appl. Prob. 23 (1986) 443–458.

[13] W. Whitt, Approximations for the GI/G/m queue, Production and Operations management 2 (2) (1993) 114–161.

[14] T. Kimura, Approximations for the delay probability in the M/G/s queue, Math. Comput. Modelling 22 (10-12) (1995) 157–165.

[15] B. N. W. Ma, J. W. Mark, Approximation of the mean queue lengyh of an M/G/c queueing system, Operations Research 43 (1) (1995) 158–165.

[16] C. L. Wang, R. W. Wolff, The M/G/c queue in light traffic, Queueing Systems 29 (1998) 17–34.

[17] W. Whitt, The impact of a heavy-tailed service-time distribution upon the M/GI/s waiting-time distribution, Queueing Systems 36 (2000) 71–87.

[18] A. Scheller-Wolf, K. Sigman, New bounds for expected delay in FIFO GI/GI/c queues, Queueing Systems 26 (1997) 169–186.

[19] D. Stoyan, Comparison methods for queues and other stochastic models, Wiley Series in Probability and Mathematical Statistics, 1983.

[20] P. Hokstad, Approximations for the M/G/m queue, Operations Research 26 (3) (1978) 510–523.

[21] A. M. Lee, P. A. Longton, Queueing processes associated with airline passenger Check-In, Oper. Res. Quart. 10 (1957) 56–71.

[22] D. Stoyan, Approximations for M/G/s queues, Mathematische Operationsforschung und Statistik 7 (4) (1976) 587–594.

[23] S. Nozaki, R. Ross, Approximations in finite capacity multiserver queues with Poisson Arrivals, Journal of Applied Probability 13 (1978) 826–834.

[24] O. J. Boxma, J. W. Cohen, N. Huffels, Approximations of the mean waiting time in an M/G/s queueing system, Operations Research 27 (6) (1979) 1115–1127.

[25] G. P. Cosmetatos, Some approximate equilibrium results for the multiserver queue M/G/r, Oper. Res. Quart. 27 (1976) 615–620.

[26] H. C. Tijms, M. H. van Hoorn, A. Federgruen, Approximations for the steady-state probabilities in the M/G/c queue, Adv. Appl. Prob. 13 (1981) 186–206.

[27] D. Burman, D. R. Smith, A light-traffic theorem for multi-server queues, Mathematics of Operations Research 8 (1) (1983) 15–24.

[28] P. Barford, A. Bestavros, A. Bradley, M. Crovella, Changes in web client access patterns: Characteristics and caching implications, WWW Journal 2 (1-2) (June 1999) 15–28.

[29] P. Barford, M. Crovella, Generating representative web workloads for network and server performance evaluation, in: Proceedings of the ACM SIGMETRICS Conference, 1998.

[30] A. Feldmann, Characteristics of TCP connection arrivals, in: Self-Similar Network Traffic and Performance Evaluation, K.Park and W.Willinger, editors, Wiley, 2000.

[31] A. Feldmann, A. C. Gilbert, W. Willinger, Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic, in: Proceedings of ACM SIGCOMM, 1998.

[32] C. J. Nuzman, I. Saniee, W. Sweldens, A. Weiss, A compound model for TCP connection arrivals, in: Proceedings of ITC Seminar on IP Traffic Modeling, Monterey, 2000.

[33] V. Paxson, S. Floyd, Wide area traffic: the failure of Poisson modeling, IEEE/ACM Transactions on Networking 3 (3) (1995) 226–244.

[34] W. S. Cleveland, D. Lin, D. X. Sun, IP packet generation: statistical models for TCP start times based on connection-rate superposition, in: Proceedings of ACM SIGMETRICS, 2000, pp. 166–177.

[35] S. B. Fredj, T. Bonalds, A. Prutiere, G. Gegnie, J. Roberts, Statistical bandwidth sharing: a study of congestion at flow level, in: Proceedings of SIGCOMM, 2001.

[36] Sprint ATL, Sprint network traffic flow traces (2002).
URL
`http://www.sprintlabs.com/Department/IP-Interworking/Monitor/`

[37] L. Kleinrock, Queueing Systems Volume II: Computer Applications, John Wiley and Sons, 1976.

[38] Y. Takahashi, An approximation formula for the mean waiting time of an $m/g/c$ queue, J. Operat. Res. Soc. Japan 20 (1977) 150–163.

[39] M. Abramowitz, I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover, 1972.