PARTITIONING CUSTOMERS INTO SERVICE GROUPS

by

Ward Whitt[1]

AT&T Labs

September 23, 1997

Revision: February 9, 1998

Management Science 45 (1999) 1579–1592

[1]Room A117, AT&T Labs, Shannon Laboratory, 180 Park Avenue, Florham Park, NJ 07932-0971; email: wow@research.att.com

Abstract

We explore the issue of when and how to partition arriving customers into service groups that will be served separately, in a first-come first-served manner, by multi-server service systems having a provision for waiting, and how to assign an appropriate number of servers to each group. We assume that customers can be classified upon arrival, so that different service groups can have different service-time distributions. We provide methodology for quantifying the tradeoff between economies of scale associated with larger systems and the benefit of having customers with shorter service times separated from other customers with longer service times, as is done in service systems with express lines. To properly quantify this tradeoff, it is important to characterize service-time distributions beyond their means. In particular, it is important to also determine the variance of the service-time distribution of each service group. Assuming Poisson arrival processes, we then can model the congestion experienced by each server group as an M/G/s queue with unlimited waiting room. We use previously developed approximations for M/G/s performance measures to quickly evaluate alternative partitions.

Keywords: queues, multi-server queues, service systems, service-system design, resource sharing, service systems with express lines

1. Introduction

In this paper we consider how we can exploit information about customers to design effective service systems. We start with an arrival stream of customers requiring service. The required service is characterized by its duration — a service time. We assume that it is possible to classify these customers according to some attributes, so that we obtain different classes with identifiable service-time distributions. We consider partitioning these customer classes into disjoint subsets that will be served separately in multi-server queues with unlimited waiting space and the first-come first-served (FCFS) service discipline. In addition to selecting the service groups, we determine how many servers to assign to each. Our goal is to efficiently meet customer performance requirements. In particular, we seek the partition and server assignment that minimizes the total number of servers used, subject to constraints on the waiting-time distribution of each class.

Of course, there are other ways to provide different levels of service to the different classes, e.g., by using a non-FCFS service discipline, such as a priority or round robin scheme, within a single facility. There also are other possible motivations for partitioning customers besides efficiently meeting performance objectives; e.g., different customers may require very different kinds of service or different customers may be geographically separated. Even when there is flexibility in partitioning, as we have defined it, there are typically other costs and benefits, e.g., see Larson (1987) and Rothkopf and Rech (1987). Nevertheless, we believe our narrow focus is useful for developing a better understanding about the performance of service systems.

As background for the analysis here, it is useful to recall what is known about the case in which all customers have the same service-time distribution. In that situation, it is known that it is more efficient to have aggregate systems, everything else being equal; e.g., see Smith and Whitt (1981) and Whitt (1992). Of course, sometimes we are constrained to use a given set of service facilities with at least one server in each; see Green and Guha (1995) for an analysis of that situation (with common service-time distributions). However, with flexibility and common service-time distributions, we should select a single aggregate system. In contrast, here we are interested in the case of *different* service-time distributions. With different service-time distributions, the service-time distributions are altered in the partitioning process. With different service-time distributions, there is a tradeoff between the economies of scale gained from larger systems and the cost of having customers with shorter service times have their quality of service degraded by customers with longer service times. Thus, there is a natural motivation for separation, as in the express checkout lines in a supermarket.

We assume that all customer classes arrive in independent Poisson processes. Thus the arrival process for any subset in the partition, being the superposition of independent Poisson processes, is also a Poisson process. Hence, we model the performance of each subset as an M/G/s service system with s servers, an unlimited waiting space and the FCFS service discipline. The problem is to form a desirable partition and assign an appropriate number of servers to each subset in the partition.

When the different classes have different service-time distributions, the service-time distribution for each subset in the partition is a mixture of the component service-time distributions. This makes the mean just the average of the component means. If the component means are quite different, though, then the subset service-time distribution will tend to be highly variable, as reflected by its squared coefficient of variation (SCV, variance divided by the square of its mean). This high variability will in turn tend to degrade the performance of the M/G/s queue for the subset.

The problem we have posed is challenging, because even though the M/G/s model is a familiar standard queueing model, it is difficult to calculate exact performance measures for it, and we want to be able to rapidly calculate performance measures for many instances of the model to do optimization. Hence, we use relatively simple approximations, as in Whitt (1992, 1993). In fact, there now is a substantial literature on the use of such queueing approximations to design and analyze complex service and manufacturing systems, often involving networks of queues; e.g., see Bitran and Dasu (1992), Buzacott (1996), Buzacott and Shanthikumar (1992, 1993), Mandelbaum and Reiman (1998) and references therein. Thus, to a large extent, the present paper should be regarded as an expository paper illustrating that general approach in a relatively simple focused context, in the same spirit as Whitt (1985), which considered the problem of determining the best order for queues in series.

Before proceeding, we want to emphasize some assumptions that we are making. First, we are exploiting only customer-class information; we are not considering dynamic assignment based on the state of the servers and queues. Second, we allow no sharing among different service groups after the partitioning is done; the arrivals from each group always go to their designated service system. In contrast, many partitioned service systems in practice, such as in supermarket check out with express lines, allow some customers to have the option of joining other queues. A form of sharing is also achieved by jockeying, i.e., moving from one queue to another, which is discussed by Rothkopf and Rech (1987). Unfortunately, separate queues with sharing tend to be difficult to analyze; see Green (1985) for an analysis of the two-group case with sharing. Results without sharing may provide lower bounds for cases in which sharing is allowed, representing the guaranteed performance for each service group in worst-case scenarios in which all other service groups are in overload, and thus provide no opportunity for sharing. (We are assuming that sharing will be beneficial if it is appropriate controlled.)

Third, we assume that the customer service-time distributions are unaffected by the partitioning, but in general that need not be the case. Combining classes might actually make it more difficult to provide service, e.g., because servers may need different skills to serve different classes. This variation might be analyzed within our scheme by introducing parameters η_{ij} for each pair of classes (i, j) with $i \neq j$. We could then have each service time of a customer of class i multiplied by η_{ij} if classes i and j belong to the same subset in the partition. This would cause the mean to be multiplied by η_{ij} but leave the SCV unchanged. This modification would require recalculation of the two service-time moments for the subsets in the partition, but we still could use the M/G/s analysis described here. More general variants take us out of the M/G/s framework, and thus remain to be considered.

Here is how the rest of this paper is organized: In Section 2 we illustrate the advantages of separating disparate classes by considering a numerical example with three classes having very different service-time distributions. In Section 3 we illustrate the potential advantage of partitioning according to service requirement by considering a numerical example with a Pareto service-time distribution. We use the Pareto-distribution because it is a typical heavy-tailed distribution, with occasional very large values. Such heavy-tailed distributions have been found to occur in computer and communication systems; e.g., see Crovella and Bestavros (1996), Willinger, Taqqu, Sherman and Wilson (1995) and references therein. When there are customers with extra long service times, there is a natural motivation to separate them from other customers in some way. We split the Pareto distribution into five subintervals. We show there that the partition may well be preferred to one aggregate system. For both examples in Sections 2 and 3, an analysis using M/M/s models (ignoring the service-time distribution beyond its mean) leads to the incorrect conclusion that a single aggregate service group is optimal.

In the remaining sections we specify the analysis techniques. In Section 4 we review the simple approximations for M/G/s performance measures that we use. In Section 5 we indicate how to calculate the parameters of the subset service-time distributions when we partition according to service times. In Section 6 we indicate how to calculate service-time parameters when we aggregate classes. In Section 7 we indicate how we can select a reasonable initial number of servers for an M/G/s system, after which we can tune for improvement. In Section 8 we briefly discuss other model variants; e.g., we point out that the situation is very different when there is no provision for waiting. Finally, in Section 9 we state our conclusions.

2. A Multi-Class Example

In this section we give a numerical example illustrating how to study the possible partitioning of classes into service groups. We let the classes have quite different service-time distributions to demonstrate that aggregation is not always good. In particular, we consider three classes of M/M input, each with common offered load 10. The offered load for class i is the arrival rate λ_i times the mean service time $ES_i = m_{i1}$. Classes 1, 2 and 3 have arrival-rate and mean-service-time pairs (λ_i, m_{i1}) of $(10.0, 1.0)$, $(1.0, 10.0)$ and $(0.1, 100.0)$, respectively. Each class separately arrives according to a Poisson process and has exponential service times. Thus each class separately yields an M/M/s queue when we specify the number of servers, but the expected service times are very different.

We consider all possible aggregations of the classes, namely, the subsets $\{1,2\}$, $\{1,3\}$, $\{2,3\}$ and $\{1,2,3\}$ as well as the classes separately. The arrival rates and offered loads of the subgroups are just the sums of the component arrival rates and offered loads. However, the aggregated subgroups differ qualitatively from the single classes because the service-time distributions are no longer exponential. Instead, the service-time distributions of the aggregated subgroups are mixtures of exponentials (hyperexponential distributions) with SCVs greater than 1. (See Section 6.) The penalty for aggregation is initially quantified by the service-time SCV. The service-time SCVs for classes $\{1,2\}$, $\{1,3\}$, $\{2,3\}$ and $\{1,2,3\}$ are 5.05, 50.0, 5.05 and 26.4, respectively. Consistent with intuition, from these SCVs, we see that the two-class service group $\{1,3\}$ should not be as attractive as the other two-class service groups $\{1,2\}$ and $\{2,3\}$.

We use a scheme discussed in Section 7 to initially specify the number of servers. In particular, in each case we let s be approximately $\omega + \sqrt{\omega}$, where ω is the offered lead. Thus, for each class separately we let $s = 13$; for the two-class subgroups we let $s = 25$; and for the entire three-class set we let $s = 36$. This initial server-assignment rule exhibits the economy of scale. Since more servers are used with smaller groups (relatively), we also consider the three-class set with 39 servers, which is the sum of the separate numbers assigned to the separate classes.

We display performance measures calculated according to Section 4 in Table 1. From

	classes in server group							
	1	2	3	1, 2	1, 3	2, 3	1, 2, 3	
λ	10.0	1.0	0.1	11.0	10.1	1.1	11.1	
ES	1.0	10.0	100.0	1.818	1.980	18.18	2.703	
$\omega = \lambda ES$	10.0	10.0	10.0	20.0	20.0	20.0	30.0	
s	13	13	13	25	25	25	36	39
$\rho = \omega/s$	0.7692	0.7692	0.7692	0.800	0.800	0.800	0.833	0.769
$c_s^2 = SCV(S)$	1.0	1.0	1.0	5.05	50.0	5.05	26.4	
$P(W > 0)$	0.324	0.324	0.324	0.250	0.250	0.250	0.250	0.124
$E(W\|W > 0)$	0.333	3.33	33.3	1.10	10.1	11.0	6.17	4.11
EW	0.108	1.08	10.8	0.28	2.53	2.75	1.54	0.51
$P(W > 1)$	0.016	0.240	0.315	0.101	0.227	0.228	0.212	0.098
$P(W > 10)$	3.0×10^{-13}	0.016	0.240	0.00011	0.092	0.101	0.049	0.011
$P(W > 100)$	1.6×10^{-131}	3.0×10^{-13}	0.016	8.2×10^{-39}	1.3×10^{-5}	2.8×10^{-5}	2.3×10^{-6}	3.5×10^{-12}

Table 1: Performance measures for the three classes separately and all possible aggregated subsets in the multi-class example in Section 2.

Table 1, we see that the mean wait is about 10% of the mean service time for each class separately. Also the probability that the wait exceeds one mean service time, $P(W > ES)$ is 0.016 for each class separately. In contrast, these performance measures degrade substantially for the class with the shorter service times after aggregation. Consistent with intuition, the performance for service group $\{1, 3\}$ is particularly bad. The full aggregate service group containing classes $\{1, 2, 3\}$ performs better with 39 servers than 36, but in both cases the performance for class 1 is significantly worse than the performance for class 1 separately.

In contrast to the mean EW and the tail probability $P(W > ES)$, the probability of delay $P(W > 0)$ improves with aggregation in Table 1. The reason is that the probability of delay in an M/G/s model tends to be insensitive to the service-time distribution beyond its mean. Indeed, the approximation we use has this insensitivity property; see (4.3). Thus, when the performance criterion is probability of delay, aggregation tends to be good.

When the customer classes are specified at the outset, it is natural to formulate our design problem as an optimization problem. The goal can be to minimize the total number of servers used, while requiring that each class meet a specified performance requirement, e.g., the steady-state probability that a class-i customer has to wait more than d_i should be less than or equal to p_i for all i. These requirements might well not be identical for all classes. It is natural to measure the waiting time (before beginning service) relative to the service time or expected service time; i.e., customers with longer service times should be able to tolerate longer waiting times (although it is easy to think of exceptions). The alternatives that must be considered in the optimization problem are the possible partitions that can be used and the numbers of servers that are used in the subsets. When there are not many classes, this optimization problem can be easily solved with the aid of the approximations by evaluating all (reasonable) alternatives.

Given a specification of performance requirements, e.g., delay constraints $P(W_i > d_i) \leq p_i$ for class i, $i = 1, 2, 3$, we can easily find the minimum number of servers satisfying all the constraints (exploiting the best aggregation scheme) by doing calculations for a range of s for each possible service group. (By "calculations" we mean the M/G/s approximations specified in Section 4.) Starting from the initial s above, we can increase or decrease s by 1 until the constraint is just met. For example, if $d_i = ES_i$ and $p_i = 0.020$, then having the three separate classes is optimal. For the classes separately, as in the first three columns of Table 1, with 12, 13 and 14 servers, $P(W_i > ES_i) \approx 0.064$, 0.016 and 0.0041, respectively. Hence, a unit change in the number of servers makes a big change in the performance measures. The total number of servers required for the aggregate system to have $P(W > 1) \leq 0.020$ is 46, seven more than with the three separate classes.

This example illustrates the importance of considering the service-time distribution beyond its mean. If instead we assumed that the aggregate system were an M/M/s system, then the service-time SCV would be 1 instead of 26.4. Then the M/M/s model underestimates the correct mean by approximately by a factor of 13.7. Using the M/M/s model for the aggregate system, we would deduce that we only needed 37 servers in order to have $P(W > 1) \leq 0.020$ (then $P(W > 1) \approx 0.015$). We would also wrongly conclude that the aggregate system is better than the separate classes.

We believe that it usually should be reasonable to use class-dependent waiting-time criteria, as above. Everything else being equal, it is natural to relate the delay to the mean service time of the class, as above. However, we might also be interested in overall average behavior. From Table 1, we see that in this example the overall average expected wait for the three classes, each with 13 servers, is also less than the expected wait in the single combined system with 39 servers. In particular,

$$\frac{10(0.108) + 1(1.08) + 0.1(10.8)}{10 + 1 + 0.1} = \frac{3.24}{11.1} = 0.29 < 0.51 \ .$$

The queueing formulas show that there are strong economies of scale motivating larger aggregate service groups, but this example illustrates that in some circumstances (e.g., with very different service requirements) partitioning may be desirable.

3. A Distribution-Splitting Example

In the most favorable circumstance, we may be able to learn (or closely approximate) each customer's required service time upon arrival. Then we can consider classifying the customers upon arrival according to their required service times. We can then partition the positive halfline into finitely many disjoint subsets and let customers with service times in a common subset all belong to the same service group. If the subsets are subintervals, then partitioning customers according to service times tends to reduce variability; i.e., the variability within each class usually will be less than the overall variability. (But see Remark 5.1.)

Since there are (infinitely) many possible ways to partition service times into subintervals, there are (infinitely) many possible designs. The problem can be simplified by exploiting two basic principles: First, the advantage of partitioning usually stems from separating short service times from long ones. Thus, it is natural for all the partition subsets to be subintervals.

The second principle is that we should not expect to have a very large number of subsets in the partition, because a large number tends to violate the efficiency of large scale. Thus it is natural to only look for and then compare the best (or good) partitions of size 2, 3, 4, and 5, say. For example, it is natural to consider giving special protection to one class with the shortest service times; e.g., express lanes in supermarkets. It is also natural to consider protecting the majority of the customers from the customers with the largest service times; e.g., large file transfers over the internet. If only these two objectives are desired, then only three classes are needed. It is not difficult to examine candidate pairs of boundary points within a specified ordering, by essentially employing exhaustive search.

We now illustrate the service-time partitioning by considering a numerical example. We start with a single M/G input consisting of a Poisson arrival process having arrival rate $\lambda = 100$ and a Pareto service-time distribution. Let G be the service-time cdf and let $G^c(t) = 1 - G(t)$ be the associated complementary cdf (ccdf). The Pareto ccdf is

$$G^c(t) = \frac{1}{(1+bt)^{\alpha}}, \quad t \geq 0 \; , \tag{3.1}$$

and the associated density is

$$g(t) = \frac{b\alpha}{(1+bt)^{\alpha+1}}, \quad t \geq 0 \; . \tag{3.2}$$

A Pareto distribution is a good candidate model for relatively more variable (heavy-tailed) service-time distributions. The mean is infinite if $\alpha \leq 1$. If $\alpha > 1$, then the mean is $(\alpha - 1)^{-1}$.

If $1 < \alpha \leq 2$, then the variance is infinite. If $\alpha > 2$, then the SCV is

$$c_s^2 = 1 + 2\left[\frac{(\alpha-1)^2}{\alpha-2} - \alpha\right] . \tag{3.3}$$

We consider the specific Pareto ccdf in (3.1) with $\alpha = 2.1$ and $\beta = (1.1)^{-1}$, so that it has mean 1 and SCV $c_s^2 = 21$ (but infinite third moment). The offered load is 100, so that the total number of servers must be at least 101 in order to have a stable system. Using the initial sizing formula in Section 7, we initially let $s = 100 + \sqrt{100} = 110$ in the single-group partition. This yields a probability of delay of $P(W > 0) = 0.2838$, a conditional mean delay of $E(W|W > 0) = 1.21$ and a mean delay of $EW = 0.3433$. The median of the chosen Pareto distribution is 0.43, so that 50% of the service times are less than 0.43. Indeed, the conditional mean service time restricted to the interval $[0, 0.43]$ is $ES = 0.179$. The conditional mean wait given that it is positive, of 1.21 is about 6.8 times this mean; the actual mean wait 0.343 is about 2 times the mean service time. These mean waits might be judged too large for the customers with such short service requirements. Thus, assuming that we know customer service requirements upon arrival, we might attempt to make waiting times more proportional to service times by partitioning the customers according to their service-time requirements.

Here we consider partitioning the customers into five subsets using the boundary points 0.43, 2.2, 10 and 1000. The first two boundary points were chosen to be the 50^{th} and 90^{th} percentiles of the service-time distribution, while the last two boundary points were chosen to be one and three orders of magnitude larger than the overall mean 1, respectively. (The heavy tail of the Pareto distribution makes large boundary points reasonable.) In particular, from formula (5.8), we find that the probabilities that a service time falls into the interval $(0, 0.43)$, $(0.43, 2.2)$, $(2.2, 10)$, $(10, 1000)$ and $(1000, \infty)$ are 0.50, 0.40, 0.092, 0.0078 and 0.61×10^{-6}, respectively.

For each subinterval, we calculate the conditional mean and second moment given that the service time falls in the subinterval (using formulas (5.9) and (5.10)), thus obtaining the subinterval mean and second moment. The subinterval SCV is then obtained in the usual way. We display these results in Table 2. Note that these subgroup service-time SCVs are indeed much smaller than the original overall Pareto SCV of $c_s^2 = 21$.

Given the calculated characteristics for each subinterval, we can treat each subinterval as a separate independent M/G/s queue. The arrival rate is 100 times the subinterval probability. The offered load, say ω_i, is the arrival rate times the mean service time. Using the initial-sizing formula in Section 7, we let the number of servers in each case be the least integer greater than

$\omega_i + \sqrt{\omega_i}$. We regard this value as an initial trial value that can be refined as needed. Finally, the traffic intensity ρ_i is just the offered load divided by the number of servers, i.e., $\rho_i = \omega_i / s_i$. We display all these results in Table 2.

Next we describe the performance of each separate M/G/s queue using the formulas in Section 4. From Table 2, we see that the mean wait EW_i for each class i is substantially less than the mean service time of that subclass. We also calculate the probability that the waiting

	service-time intervals				
	$(0, 0.43)$	$(0.43, 2.2)$	$(2.2, 10)$	$(10, 1000)$	$(1000, \infty)$
probability	0.5000	0.4004	0.0918	0.0078	0.00000061
subgroup mean m_{i1}	0.1787	0.9811	3.935	19.94	1910
SCV c_{si}^2	0.463	0.218	0.193	1.25	4.75
λ_i	49.99	40.04	9.18	0.78	0.000061
ω_i	8.94	39.28	36.12	15.54	0.117
s_i	12	46	42	20	1
ρ_i	0.745	0.853	0.864	0.778	0.117
$P(W_i > 0)$	0.299	0.251	0.298	0.252	0.117
$E(W_i \mid W_i > 0)$	0.043	0.088	0.411	5.03	6215
EW_i	0.0128	0.0222	0.122	1.27	724
$P(W_i > ES_i)$	0.0045	0.000004	0.000021	0.0048	0.086

Table 2: Service-time characteristics and M/G/s performance measures when the original Pareto service times are split into five subgroups.

time exceeds the mean service time of that class, using approximation (4.11) in Section 4. For all classes, $P(W_i > ES_i)$ is consistently small.

We conclude this section with a word of caution. The approximations in Section 4 are not intended for the case of heavy-tailed service-time distributions. The approximations can be very optimistic for heavy-tailed service-time distributions, so the approximations for any service group containing the last class should be used with caution. Note, however, that this caution does not apply to the other classes, which all have bounded service times as a consequence of the partitioning. We discuss the heavy-tailed case further at the end of Section 4.

Remark 3.1. To illustrate the difficulty with heavy-tailed service-time distributions, we could consider a Pareto distribution with $\alpha < 2$. (The Pareto service-time distribution in the example we have considered has finite variance since $\alpha = 2.1 > 2$.) Similar results hold if the service-time distribution has infinite variance or even infinite mean. When the service-time distribution has finite mean but infinite variance (when $1 < \alpha \leq 2$), the service-time variance is finite for all subclasses but the last because of truncation. The service-time distribution for the last class then has finite mean and infinite second moment. In the example here with one server assigned to the last class, we then have $P(W > 0) = \rho < 1$, but $EW = \infty$. When the mean service-time is infinite for the last class (when $\alpha \leq 1$), the waiting times for that class diverge to $+\infty$. However, the other classes remain well behaved. Clearly, the splitting may well be deemed even more important in these cases. But approximations for the waiting-time cdf of a service group including the last class should be regarded as optimistic.

Now we consider what happens if we aggregate some of the subgroups. First, we consider combining the last two subgroups. We keep the total number of servers the same at 21. If we group the last two classes together, then the new service time has mean 20.09 and SCV 5.29. Note that, compared to the (10, 1000) class, the mean has gone up only slightly from 19.94, but the SCV has increased significantly from 1.25. (The SCV is even bigger than it was for the highest group.) The M/G/s performance measures for the new combined class are $P(W > 0) = 0.1921$, $E(W|W > 0) = 11.85$, $EW = 2.28$ and $P(W > ES) = 0.035$. This combination might be judged acceptable, but the performance becomes degraded for the customers in the (10, 1000) subgroup.

Next, we consider aggregating all the subgroups. If we keep the same numbers of servers assigned to the subgroups, then we obtain 121 servers instead of 110. This should not be surprising, because the $\omega + \sqrt{\omega}$ algorithm should produce fewer extra servers with one large group than with five subgroups. However, it still remains to examine the performance of the original system when $s = 121$. When $s = 121$, $P(W > 0) = 0.062$, $E(W|W > 0) = 0.634$, $EW = 0.0393$, $P(W > 0.1787) = 0.0438$, and $P(W > 0.9811) = .0094$. The delay probability is clearly better than with the partition, as it must be using approximation (4.3), but the conditional mean wait is worse for the first three subgroups, and much worse for the first two. The overall mean EW is worse for the first two subgroups, and much more for the first one. The tail probabilities $P(W > ES_i)$ are much worse for the first two subgroups as well. Hence, even with all 121 servers, performance in the single aggregated system might be considered far inferior to performance in the separate groups for the first two groups.

Finally, we can clearly see here that an M/M/s model fails to adequately describe the performance. In this case, the mean EW would be underestimated by a factor of 11 in the aggregate system, and overestimated somewhat for the first three service groups. Moreover, we would incorrectly conclude that the aggregate system must be better.

Remark 3.2. In this section we did not specify a specific optimization problem, but it is easy to do so. We could give criteria for forming the service groups. We might specify upper and/or lower bounds on the proportion of the total arrival rate that can be in each group. As in Section 2, we can specify constraints on the delay distributions, which could be expressed in terms of the mean service time of that service group, e.g., $P(W_i > cES_i) \leq p$ for all i. As in Section 2, we can minimize the total number of servers required subject to those constraints. The optimal solution can be found by searching over the number of service groups (e.g., initially considering up to three, and then considering higher numbers only if three is better than fewer), the service group boundaries (e.g., expressed as percentiles of the known service-time distribution, in units of 0.02, say, with small last "tail" groups allowed, in units of 10^{-k}, say) and the number of servers in each service group. For the number of servers in each group, we can start with the initial value $\omega + \sqrt{\omega}$ and then increase or decrease by one until finding the point that the constraint is just met.

4. Review of M/G/s Approximations

In this section we review the approximations for the performance measures of M/G/s systems. The approximations make it possible to quickly determine the approximate performance of an M/G/s system, so that we can quickly evaluate possible partition schemes. We use the steady-state probability of having to wait, $P(W > 0)$, and the steady-state conditional expected wait given that the customer must wait, $E(W|W > 0)$. The product of these two is of course the mean steady-state wait itself, EW. We also use the steady-state tail probability $P(W > t)$. Relevant choices of t typically depend on the mean service time, here denoted by ES.

The basic model parameters are the number of servers s, the arrival rate λ and the service-time cdf G with k^{th} moments m_k, $k \geq 1$. The traffic intensity is $\rho = \lambda m_1/s$; we assume that $\rho < 1$, so that a proper steady state exists. This condition puts an obvious lower bound on the number of servers in each service group. The service-time SCV is $c_s^2 = (m_2 - m_1^2)/m_1^2$.

The approximations have been quite extensively studied, e.g., see Whitt (1993), so we do not address their accuracy here. We could use simulation or more involved numerical algorithms, as in de Smit (1983), Seelen (1986) and Bertsimas (1988), to more accurately calculate the exact performance measures, but we contend that it is usually not necessary to do so, because the approximation accuracy tends to be adequate and the approximations are much easier to use and understand. The adequacy of approximation accuracy depends in part on the intended application to determine the required number of servers. As we saw in Section 2, a small change in the number of servers (e.g., by one) typically produces a significant change in the waiting-time performance measures. This means that the approximation error has only a small impact on the decision. Moreover, the approximation accuracy is often much better than our knowledge of the underlying model parameters (arrival rates and service-time distributions). If, however, greater accuracy is deemed necessary, then one of the alternative exact numerical algorithms can be used in place of the approximations here.

We now specify the proposed approximations. First recall that the exact conditional wait for M/M/s is

$$E(W(M/M/s)|W(M/M/s)>0)) = \frac{m_1}{s(1-\rho)} \ , \tag{4.1}$$

which is easy to see because an M/M/s system behaves like an M/M/1 system with service rate s/m_1 when all s servers are busy. As in Whitt (1993) and elsewhere, we approximate the conditional M/G/s wait by

$$\begin{aligned} E(W(M/G/s)|W(M/G/s)>0) &\approx \frac{(1+c_s^2)}{2} E(W(M/M/s)|W(M/M/s)>0) \\ &= \frac{(1+c_s^2)m_1}{2s(1-\rho)} \ , \end{aligned} \tag{4.2}$$

which is exact for $s=1$.

Following Whitt (1993) and references cited there, we approximate the probability of delay in an M/G/s system by the probability of delay in an M/M/s system with the same traffic intensity ρ, i.e.,

$$P(W(M/G/s)>0) \approx P(W(M/M/s)>0) \tag{4.3}$$

where

$$P(W(M/M/s)>0) = [(s\rho)^s/s\ (1-\rho)]\zeta \tag{4.4}$$

with

$$\zeta = \left[\frac{(s\rho)^s}{s!(1-\rho)} + \sum_{k=0}^{s-1} \frac{(s\rho)^k}{k!} \right]^{-1} \ . \tag{4.5}$$

Algorithms are easily constructed to compute the exact M/M/s delay probability in (4.4). However, we also propose the more elementary Sakasegawa (1977) approximation

$$P(W(M/M/s)>0) \approx \rho^{\sqrt{2(s+1)}-1} \ . \tag{4.6}$$

When we combine (4.2) and (4.3), we obtain the classic Lee and Longton (1959) approximation formula for the mean, i.e.,

$$EW(M/G/s) \approx \frac{(1+c_s^2)}{2} EW(M/M/s) \approx \frac{(1+c_s^2)m_1}{2s(1-\rho)} P(W(M/M/s)>0 \ , \tag{4.7}$$

which we complete either with an exact calculation or approximation (4.6).

We may know roughly what the probability of delay will be; e.g., we might have $P(W > 0) \approx 0.25$. Then we can obtain an explicit back-of-the-envelope approximation by substituting that approximation into (4.7). A simple heavy-traffic approximation is obtained by letting $P(W(M/M/s) > 0) \approx 1$ in (4.7) or, equivalently, $EW \approx E(W|W > 0)$, which amounts to using (4.2). Then for fixed ρ, there is a clear tradeoff between s (scale) on the one hand and $(1+c_s^2)m_1$ (a combination of mean and variability of the service-time distribution) on the other hand.

Remark 4.1. As indicated above, we can do a heavy-traffic analysis to quickly see the benefits of service-time splitting. Suppose that we allocate servers proportional to the offered load, so that $\rho_i = \rho$ for all i. Since $\rho = \lambda m_1/s$,

$$s_i = \frac{\lambda_i m_{i1}}{\rho_i} = \frac{\lambda_i m_{i1}}{\rho} = \left(\frac{\lambda_i m_{i1}}{\lambda m_1}\right) s \; . \tag{4.8}$$

Then, by (4.2),

$$EW \approx \frac{m_1(1+c_s^2)}{2s(1-\rho)} \tag{4.9}$$

and

$$EW_i \approx \frac{m_{i1}(1+c_{si}^2)}{2s_i(1-\rho_i)} = \frac{\lambda m_1(1+c_{si}^2)}{2\lambda_i s(1-\rho)} = \frac{\lambda(1+c_{si}^2)}{\lambda_i(1+c_s^2)} EW \tag{4.10}$$

for EW in (4.9). Hence, we should have $EW_i < EW$ if and only if $(1+c_{si}^2)/\lambda_i < (1+c_s^2)/\lambda$. If the original classes each have deterministic service times, then this condition becomes $1+c_s^2 > \lambda/\lambda_i$. This simple analysis shows that important role played by service-time variability, as approximately described by the SCVs c_s^2 and c_{si}^2.

We can approximate the tail probability roughly by assuming that the conditional delay is exponential, i.e.,

$$P(W > t) \approx P(W > 0)e^{-t/E(W|W>0)} \; , \quad t > 0 \; . \tag{4.11}$$

Approximation (4.11) is exact for M/M/s, but not more generally.

As mentioned in Section 3, approximation (4.11) is likely to be very optimistic for heavy-tailed service-time distributions. To understand how optimistic approximation (4.11) can be, we give an approximation for $P(W > t)$ in the single-server case (M/G/1) when the service-time ccdf G^c has a heavy tail, which is asymptotically correct (in ratio) as $t \to \infty$. The approximation is

$$P(W > t) \approx \frac{\rho}{1-\rho} \int_{t/ES}^{\infty} G^c(u)du \; ; \tag{4.12}$$

see Abate, Choudhury and Whitt (1994). Formula (4.11) differs dramatically from (4.11); it shows that $P(W > t)$ inherits the heavy-tail property of G^c.

For the M/G/s system with a heavy-tailed service-time ccdf and $s > 1$, we know no good simple approximation for $P(W > t)$. However, it is known that $P(W > t)$ still inherits the heavy-tail property from the service-time ccdf G^c; see Whitt (1998). The M/G/s algorithms by de Smit (1983), Seelen (1986) and Bertsimas (1988) can be applied after approximating the service-time distribution by a light-tailed distribution such as a hyperexponential distribution or a phase-type distribution; see Asmussen, Nerman and Olsson (1996), Feldmann and Whitt (1998) and Harris and Marchal (1999).

5. Splitting by Service Times

Suppose that we are given a single M/G input with arrival rate λ and service-time cdf G. We can create m classes by classifying customers according to their service times, which we assume can be learned upon arrival. We use $m-1$ numbers $x_1, x_2, \ldots, x_{m-1}$ with $0 < x_1 < x_2 < \cdots < x_{m-1}$. Let $x_0 = 0$ and $x_m = \infty$. We say that an arrival belongs to class i, $2 \le i \le m-1$, if its service time falls in the interval $(x_{i-1}, x_i]$. For class 1 the interval is $[0, x_1]$; for class m the interval is (x_{m-1}, ∞).

Since the service times are assumed to be independent and identically distributed (i.i.d), this classification scheme partitions the original Poisson arrival process into m independent Poisson arrival processes. Thus one M/G input has been decomposed into m independent M/G inputs (without yet specifying the numbers of servers).

The arrival rate of class i is thus

$$\lambda_i = \lambda[G(x_i) - G(x_{i-1})] \tag{5.1}$$

(regarding $G(0)$ as $G(0-) = 0$) and the associated service-time cdf is

$$G_i(x) = \begin{cases} 0, & x \le x_{i-1} \\ \frac{G(x)}{G(x_i) - G(x_{i-1})}, & x_{i-1} \le x \le x_i \\ 1, & x > x_i \ . \end{cases} \tag{5.2}$$

The k^{th} moment of G_i is

$$m_{ik} = \frac{1}{[G(x_i) - G(x_{i-1})]} \int_{x_{i-1}}^{x_i} x^k dG(x) \ . \tag{5.3}$$

It is significant that the moments of the split cdf's can be computed in practice, as we now illustrate.

Example 5.1. (exponential distributions). Suppose that the cdf G is exponential with mean μ^{-1}, so that the density is

$$g(t) = \mu e^{-\mu t}, \quad t \geq 0 . \tag{5.4}$$

If G_i is G restricted to the interval $(x_{i-1}, x_i]$, then

$$G(x_i) - G(x_{i-1}) = e^{-\mu x_{i-1}} - e^{-\mu x_i} . \tag{5.5}$$

The first two moments of G_i are then

$$m_{i1} = \frac{1}{\mu[G(x_i) - G(x_{i-1})]}(e^{-\mu x_{i-1}}(1 + \mu x_{i-1}) - e^{-\mu x_i}(1 + \mu x_i)) \tag{5.6}$$

and

$$m_{i2} = \frac{1}{\mu^2[G(x_i) - G(x_{i-1})]}(e^{-\mu x_{i-1}}(2 + 2\mu x_{i-1} + \mu^2 x_{i-1}^2) - e^{-\mu x_i}(2 + 2\mu x_i + \mu^2 x_i^2)) . \tag{5.7}$$

Example 5.2. (Pareto distributions). Now consider the Pareto ccdf in (3.1) used in Section 3. If G_i is G restricted to the interval $(x_{i-1}, x_i]$, then

$$G(x_i) - G(x_{i-1}) = G^c(x_{i-1}) - G^c(x_i) = \frac{1}{(1 + bx_{i-1})^\alpha} - \frac{1}{(1 + bx_i)^\alpha} . \tag{5.8}$$

We can apply formula (5.3) to calculate the first two moments of G_i. They are

$$\begin{aligned} m_{i1} &= \frac{\alpha}{b[G(x_i) - G(x_{i-1})]}\left(\frac{(1 + bx_i)^{1-\alpha} - (1 + bx_{i-1})^{1-\alpha}}{1 - \alpha}\right. \\ &\quad + \left.\frac{(1 + bx_i)^{-\alpha} - (1 + bx_{i-1})^{-\alpha}}{\alpha}\right) \end{aligned} \tag{5.9}$$

and

$$\begin{aligned} m_{i2} &= \frac{\alpha}{b^2[G(x_i) - G(x_{i-1})]}\left(\frac{(1 + bx_i)^{2-\alpha} - (1 + bx_{i-1})^{2-\alpha}}{2 - \alpha}\right. \\ &\quad + \left.\frac{2(1 + bx_{i-1})^{1-\alpha} - 2(1 + bx_i)^{1-\alpha}}{1 - \alpha} + \frac{(1 + bx_{i-1})^{-\alpha} - (1 + bx_i)^{-\alpha}}{\alpha}\right) . \end{aligned} \tag{5.10}$$

Remark 5.1. When we split service times, we expect to have $c_{si}^2 < c_s^2$, but that need not be the case. First, if G is uniform on $[0, x]$, then G_i is uniform on (x_{i-1}, x_i) and $c_{si}^2 < c_{s1}^2 = c_s^2 = 1/3$. Second, suppose that G assigns probabilities $\epsilon/2$, $\epsilon/2$ and $1 - \epsilon$ to 0, x_1 and $x_1 + \delta$. Then $c_{s1}^2 = 1$ for all $\epsilon > 0$ and $\delta > 0$, while $c_s^2 \to 0$ as $\epsilon \to 0$, so that we can have $c_{s1}^2 >> c_s^2$.

Remark 5.2. Although the approximations for the M/G/s queue in Section 4 depend on the service-time distribution only through its first two moments, the two moments for each class obtained by splitting the service-time distribution G depend on the full distribution of G (beyond its first two moments).

6. Aggregation

Suppose that we are given m independent M/G inputs with arrival rates λ_i and service-time cdf's G_i having k^{th} moments m_{ik}, $1 \le i \le m$. Then the m classes can be combined (aggregated) into a single M/G input with arrival rate the sum of the component arrival rates, i.e.,

$$\lambda = \sum_{i=1}^{m} \lambda_i \tag{6.1}$$

and service-time cdf a mixture of the component cdf's, i.e.,

$$G(t) = \sum_{i=1}^{m} (\lambda_i/\lambda) G_i(t), \quad t \ge 0 \ , \tag{6.2}$$

having moments

$$m_k = \sum_{i=1}^{m} (\lambda_i/\lambda) m_{ik} \ . \tag{6.3}$$

Thus the aggregate SCV is

$$c_s^2 = \frac{\Sigma \lambda_i (1 + c_{si}^2) m_{i1}^2}{\lambda m_1^2} - 1 \ . \tag{6.4}$$

It should be evident that if a single M/G input is split by service times as described in Section 3 and then recombined, we get the original M/G input characterized by λ and G back again.

7. Initial Numbers of Servers

In this section we indicate how to initially select the number of servers in any candidate M/G/s system. Our idea is to use an infinite-server approximation, as in Section 2.3 of Whitt (1992) or in Jennings, Mandelbaum, Massey and Whitt (1996). In the associated M/G/∞ system with the same M/G input, the steady-state number of busy servers has a Poisson distribution with mean (and thus also variance) equal to the offered load (product of arrival rate and mean service time, say ω. The Poisson distribution can then be approximated by a normal distribution. We thus let the number of servers be the least integer greater than or equal to $\omega + c\sqrt{\omega}$, which is c standard deviations above the mean. A reasonable value of the constant is often $c = 1$; and we will use it. Then the number of servers is

$$s = \lceil \omega + \sqrt{\omega} \rceil \ . \tag{7.1}$$

A rough estimate (lower bound) for the probability of delay is then

$$P(W > 0) \approx P(N(0,1) > 1) = \Phi^c(1) = 0.16 \ , \tag{7.2}$$

where $N(a,b)$ denotes a normal random variables with mean a and variance b, and Φ^c is the complementary cdf of $N(0,1)$, i.e., $\Phi^c(x) = P(N(0,1) > x)$. This choice tends to keep the waiting time low with the servers well utilized. Of course, the number of standard deviations above the mean and/or the resulting number of servers can be further adjusted as needed.

8. Other Model Variants

So far, we have considered service systems with unlimited waiting space. A very different situation occurs when there is no waiting space at all. The steady-state number of busy servers in an M/G/s/0 loss model has the insensitivity property; i.e., the steady-state distribution of the number of busy servers depends on the service-time distribution only through its mean. Thus, the steady-state distribution in the M/G/s/0 model coincides with the (Erlang B) steady-state distribution in the M/M/s/0 model with an exponential service-time distribution having the same mean. Thus, the full aggregated system is always more efficient for loss systems, by Smith and Whitt (1981).

Similarly, if there is extra waiting space, but delays are to be kept minimal, then it is natural to use the M/G/∞ model as an approximation, which also has the insensitivity property. Hence, if our goal can be expressed in terms of the distribution of the number of busy servers in the M/G/∞ model, then we should again prefer the aggregate system.

Even for the M/G/s delay model, our approximation for the probability of experiencing any wait in (4.3) has the insensitivity property. Hence, if our performance criterion were expressed in terms of the probability of experiencing any wait, then we also should prefer the aggregate system. In contrast, separation can become important for the delays, because the service-time distribution beyond its mean (as described by the SCV) then matters, as we have seen.

So far, we have considered a stationary model. However, in many circumstances it is more appropriate to consider a nonstationary model. For example, we could assume a nonhomogeneous Poisson arrival process, denoted by M_t, for each customer class. It is important to note that the insensitivity in the M/G/s/0 and M/G/∞ models is lost when the arrival process becomes M_t; see Davis, Massey and Whitt (1995). The added complexity caused by the nonstationarity makes it natural to consider the $M_t/G/\infty$ model as an approximation. Since insensitivity no longer holds, full aggregation is not necessarily most efficient. Partitioning in this nonstationary setting can also be conveniently analyzed because the partitioning of nonhomogeneous Poisson processes produces again nonhomogeneous Poisson processes. Hence, all subgroups behave as $M_t/G/s$ systems. For example, the server staffing and performance calculations for each subset can be performed by applying the approximation methods in Jennings, Mandelbaum, Massey and Whitt (1996). The formula for the mean number of busy servers at time t in (6) there shows that the service-time distribution beyond the mean plays a role, i.e.,

$$m(t) = E[\lambda(t - S_e)]E[S] \ , \tag{8.1}$$

where $\lambda(t)$ is the arrival-rate function and S_e is a random variable with the service-time equilibrium-excess distribution, i.e.,

$$P(S_e \leq t) = \frac{1}{ES} \int_0^t G^c(u) du \ ; \tag{8.2}$$

also see Eick, Massey and Whitt (1993). The linear approximation

$$m(t) \approx \lambda(t - \lambda'(t)E[S_e])E[S] = \lambda(t)ES - \lambda'(t)[ES]^2 \frac{(c_s^2 + 1)}{2} \tag{8.3}$$

in (8) of Eick et al. shows the first-order effect of the service-time SCV.

So far, we have only considered Poisson arrival processes. We chose Poisson arrival processes because, with them, it is easier to make our main points, and because they are often reasonable in applications. However, we could also employ approximation methods to study the partitioning of more general (stationary) G/G inputs. In particular, we could use approximations for aggregating and splitting of arrival streams in the queueing network analyzer (QNA) in Whitt (1983) to first calculate an SCV for the arrival process of each server group and then calculate approximate performance measures. When we go to this more general setting, the arrival-process variability then also has an impact. With non-Poisson arrival processes, the partitioning problem nicely illustrates how a performance-analysis software tool such as QNA can be conveniently applied to study design problem.

It should be noted, however, that the QNA formulas for superposition (aggregation) and splitting assume independence. The independence seems reasonable for aggregation, but may fail to properly represent splitting. For aggregation, the assumed independence is among the arrival processes for the different classes to be superposed, which we have already assumed in the Poisson case.

For splitting, we assume that the class identity obtained by splitting successive arrivals are determined by independent trials. Thus, if c_a^2 is the original arrival-process SCV and p_i is the probability that each arrival belongs to class i, then the resulting approximation for the class-i SCV c_i^2 from Section 4.4 of Whitt (1983) is

$$c_i^2 = p_i c_a^2 + 1 - p_i \ , \tag{8.4}$$

which approaches the value 1 as $p_i \to 0$. Formula (8.4) is exact for renewal processes and is consistent with limits to the Poisson for more general stationary point processes. However, in applications it is possible that burstiness (high variability) may be linked to the class attributes, so that a cluster of arrivals in the original process all may tend to be associated with a common class. That means the independence condition would be violated. Moreover, as a consequence, the actual SCV's associated with the split streams should be much larger than predicted by (8.4). In such a situation it may be better to rely on measurements, as discussed in Fendick and Whitt (1989).

9. Conclusions

We have shown how to evaluate the performance costs and benefits of partitioning customers into service groups to be served in separate M/G/s FCFS systems. The possibility for doing so depends on the ability to identify customer classes and their service-time distributions. When the service-time distributions of the customer classes do not differ greatly, then greater efficiency usually can be obtained by combining the systems. On the other hand, if the service-time distributions are very different, then it may be better to partition. To see the advantage of partitioning, it is crucial to go beyond M/M/s models to M/G/s models, where the service-time distribution can be partially characterized by its mean and SCV (or, equivalently, by its first two moments). Previously established simple approximations for M/G/s performance measures make it possible to evaluate alternatives very rapidly. Afterwards, the conclusions can be confirmed by more involved numerical algorithms, computer simulations or system measurements. (It was noted that the partial characterization of the waiting-time distribution in terms of the first two moments of the service-time distribution tends to be very optimistic if the service-time distribution has a heavy tail.)

It remains to further study and compare other schemes for providing different levels of service to different customer classes, including dynamic assignment based on system state, static partitioning with forms of sharing and single service groups with various non-FCFS service disciplines, such as round robin.

References

Abate, J., G. L. Choudhury and W. Whitt, "Waiting-Time tail Probabilities in Queues with Long-Tail Service-Time Distributions, *Queueing Systems* 16 (1994), 311–338.

Asmussen, S., O. Nerman and M. Olsson, "Fitting Phase Type Distributions via the EM Algorithm," *Scand. J. Statist.* 23 (1996), 419–441.

Bertsimas, D., "An Exact FCFS Waiting Time Analysis for a General Class of G/G/s Queueing Systems," *Queueing Systems* 3 (1988), 305–320.

Bitran, G. R. and S. Dasu, "A Review of Queueing Network Models of Manufacturing Systems," *Queueing Systems* 12 (1992), 95–133.

Buzacott, J. A., "Commonalities in Reengineered Business Processes: Models and Issues," *Management Sci.* 42 (1996), 768–782.

Buzacott, J. A. and J. G. Shanthikumar, "Design of Manufacturing Systems Using Queueing Models," *Queueing Systems* 12 (1992), 135–213.

Buzacott, J. A. and J. G. Shanthikumar, *Stochastic Models of Manufacturing Systems*, Prentice-Hall, Englewood Clifs, NJ, 1993.

Crovella, M. E. and A. Bestavros, "Self-similarity in world wide web traffic — evidence and possible causes. *Proc. ACM Sigmetrics* '96, 1996, 160–169.

Davis, J. L., W. A. Massey and W. Whitt, "Sensitivity to the Service-Time Distribution in the Nonstationary Erlang Loss Model," *Management Sci.* 4 (1995), 1107–1116.

Eick, S. G., W. A. Massey and W. Whitt, "The Physics of the $\mathrm{M_t}/\mathrm{G}/\infty$ Queue," *Oper. Res.* 41 (1993), 731–742.

Feldmann, A. and W. Whitt, "Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models," *Performance Evaluation* 31 (1998), 245–279.

Fendick, K. W. and W. Whitt, "Measurements and Approximations to Describe the Offered Traffic and Predict the Average Workload in a Single-Server Queue. *Proceedings of the IEEE* 77 (1989), 171–194.

Green, L. V., "A Queueing System with General-Use and Limited-Use servers," *Operations*

Res. 33 (1985), 168–182.

Green, L. V. and D. Guha, "On the Efficiency of Imbalance in Multi-Facility Multi-Server Service Systems," *Management Sci.* 41 (1995), 179–187.

Harris, C. M. and W. G. Marchal, "Distribution Estimation using Laplace Transforms," *INFORMS J. Computing* 10 (1998), 448–458.

Jennings, O. B., A. Mandelbaum, W. A. Massey and W. Whitt, "Server Staffing to Meet Time-Varying Demand," *Management Sci.* 42 (1996), 1383–1394.

Larson, R. C., "Perspectives on Queues: Social Justice and the Psychology of Queueing," *Operations Res.* 35 (1987), 985–905.

Lee, A. M. and P. A. Longton, "Queueing Processes Associated with Airline Passengers Check-In," *Operations Research Quarterly* 10 (1959), 56–71.

Mandelbaum, A. and M. I. Reiman, "On Pooling in Queueing Networks," *Management Sci.* 44 (1998), 971–981.

Rothkopf, M. H. and P. Rech, "Perspectives on Queues: Combining Queues Is Not Always Beneficial," *Operations Research* 35 (1987), 906–909.

Sakasegawa, H. "An Approximation Formula $L_q = \alpha\beta^\rho/(1-\rho)$," *Ann. Inst. Stat. Math.* 29 (1977), 67–75.

Seelen, L. P., "An Algorithm for Ph/Ph/c Queues," *Eur. J. Oper. Res.* 23 (1986), 118–127.

de Smit, J. H. A., "A Numerical Solution for the Multi-Server Queue with Hyperexponential Service Times," *Oper. Res. Letters* 2 (1983), 217–224.

Smith, D. R. and W. Whitt, "Resource Sharing for Efficiency in Traffic Systems," *Bell System Tech. J.* 60 (1981), 39–55.

Whitt, W., "The Queueing Network Analyzer," *Bell System Tech. J.* 62 (1983), 2779–2815.

Whitt, W., "The Best Order for Queues in Series," *Management Sci.* 31 (1985), 475–487.

Whitt, W., "Understanding the Efficiency of Multi-Server Service Systems," *Management Sci.* 38 (1992), 708–723.

Whitt, W., "Approximations for the GI/G/m Queue," *Prod. and Opns. Mgmt.* 2 (1993), 114–161.

Whitt, W., "The Impact of a Heavy-Tailed Service-Time Distribution Upon the M/G/s Waiting-Time Distribution," AT&T Labs, 1998.

Willinger, W., M. S. Taqqu, R. Sherman and D. V. Wilson, "Self similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. *Proc. ACM SIGCOMM Symp. on Commun. Architectures and Protocols*, 1995, 100–113.