102682

An N Server Cutoff Priority Queue Where Customers Request a Random Number of Servers

by

Christian Schaack and Richard C. Larson

#OR 136-85 May 1985

An N Server Cutoff Priority Queue Where Arriving Customers Request a Random Number of Servers

by

Christian Schaack
and
Richard C. Larson

Operations Research Center
Massachusetts Institute of Technology

The research reported on in this paper was conducted at MIT; it was partially supported by the National Institute of Justice (grant #84-IJ-CX-0063) and by the National Science Foundation (grant #SES-8411871).

The opinions expressed in this report are those of the authors and do not necessarily reflect the views and opinions of the sponsors.

An N Server Cutoff Priority Queue Where Arriving Customers Request a Random Number of Servers

by

Christian Schaack
and
Richard C. Larson

Operations Research Center
Massachusetts Institute of Technology

ABSTRACT

Consider a multi-priority, nonpreemptive, N-server Poisson arrival queueing system. The number of servers requested by an arrival has a known probability distribution. Service times are negative exponential. In order to save available servers for higher priority customers, arriving customers of each lower priority are deliberately queued whenever the number of servers busy equals or exceeds a given priority-dependent cutoff number. A queued priority i customer enters service the instant the number of servers busy is at most the respective cutoff number of servers minus the number of servers requested (by the customer) and all higher priority queues are empty. In other words the queueing discipline is in a sense HOL by priorities, FCFS within a priority. All servers requested by a customer start service simultaneously; service completion instants are independent. We derive the priority i waiting time distribution (in transform domain) and other system statistics.

Keywords: *priority queue, random number of servers, cutoff queue.*

1 INTRODUCTION

The model described in this paper is motivated largely by applications in police and ambulance dispatching, but it applies equally well to other areas like communications systems.

In police dispatching operations, "emergency" calls frequently require sending several patrol units to the scene of an incident. The first unit(s) on the scene cannot respond effectively to the call until all response units have arrived.

The problem is further complicated by the existence of several priority levels of emergency calls. Higher priority calls must get serviced before lower priority calls; lower priority calls have to wait for service until there are a "sufficient" number of servers available and no higher priority calls backlogged.

Unfortunately it is often impossible to recall patrol units responding to low priority calls and reassign them to a real high-priority emergency, should one arise. Because of the high risk of high priority (i.e., real emergency) calls it is therefore advisable to keep a "strategic reserve" of patrol units, even when there are low priority calls backlogged, in order to respond promptly to these potential real emergencies.

In Section 2.1 of this paper we develop a realistic queueing model of a variety of dispatching procedures typically implemented in police departments. This model provides a useful tool for the planning and design of efficient dispatching protocols. It extends the applicability of most models proposed to date by overcoming some of their most important limitations. Section 2.2 reviews the literature most relevant to our model. In Section 3 we show how to derive various measures of operational performance, including the delay probability and the mean delay in queue experienced by a priority i customer. We discuss loss systems in Section 4. Extensions and variants of our model (e.g., to allow an upper and a lower bound on the number of servers required by any given arrival rather than to have every arrival request a specific number of servers) are briefly commented on in Section 5.

2.1 MODEL DESCRIPTION

In this section we provide details of the basic mathematical model, which assumes that arriving customers either enter service immediately or join a priority-specific infinite capacity FCFS queue.

Customers are assumed to arrive in a homogeneous Poisson manner to an N server queueing system, with arrival rate λ_i (customers/unit time) for priority i customers ($i=1,2,\ldots,T$). All Poisson streams operate independently. By convention, type i customers have higher priority than type j customers if $i<j$. The time any given server spends on a job is assumed to be negative exponential with mean $1/\mu$, independent of the priority of the customer or the identity of the server.

Arriving customers require a random number of servers, in the sense that an arrival of priority i requires k servers with a probability σ_{ik}, independent of anything else. All k servers requested must start service simultaneously, though they *finish service independently of each other.*

> We need to make this independence assuption for the mathematical tractability of the stochastic-server-requirements model. This assumption may or may not be a good approximation of the reality of a potential application of the model. For police dispatching, *Green and Kolesar* [1984] empirically validate this independence assumption with data from New York City (pp.30-32). They conclude that "the i.i.d. experimental model is very good for two-car jobs and reasonably good for three-car jobs".

The service discipline is assumed to be non-preemptive, in the sense that once service has begun on a given call, it cannot be interrupted until it is completed. Priority i customers requiring k servers enter service immediately upon arrival only if there are fewer than N_i-k+1 servers busy, where N_i is the server cutoff for priority i. Otherwise they are backlogged in a queue of other priority i customers; this queue is depleted in a FCFS manner, with each depletion instant corresponding to a moment of service completion (or, more precisely, a time instant when some server finishes service) arising when the next customer in queue requires k servers and precisely N_i-k+1 servers are busy. Because the service discipline is non-preemptive, we also require that the priority $i-1$ queue be empty before priority i customers are serviced (HOL). By convention, the server cutoff number for the highest priority customers is $N_1 = N$, the number of servers. By definition the server cutoff, N_i, represents the maximum number of servers that may be busy upon the instant when a priority i customer enters service. Of course, if a priority i cutomer requests k servers, we require that $k \leqq N_i$. (We also require that the cutoffs satisfy the following inequalities: $0 < N_T \leqq ... \leqq N_2 \leqq N_1 = N$.)

A proposed shorthand notation for our model is $M/M/\{N_i\}\otimes\{\underline{S}\}$, designating Markovian (Poisson) input, Markovian (negative exponential) service times, a set of server cutoffs $\{N_i\}$, and a probability matrix $\mathbf{S} \equiv (\sigma_{ik})$ for the number of servers required.

2.2 LITERATURE REVIEW

The queueing model developed in this paper provides an analytical tool of considerable flexibility for assessing the efficiency of dispatching procedures implemented in most police departments. It overcomes some of the major limitations of most models proposed to date. One of these shortcomings is that most models are are unable to take into account multiple car dispatches. Few researchers have concerned themselves with this problem. *Green* [1980] argues that in the City of New York thirty percent of the dispatches involve multiple vehicles which makes single server queueing models rather unrealistic representations of the actual operations. Another weakness of most models used for police dispatching (and indeed of most dispatching centers' operational protocols) is that they hardly ever consider holding patrol cars in reserve for potential emergencies; such a strategy would prevent a critical shortage of resources when they are needed most. That particular problem is addressed in *Taylor and Templeton* [1980] , *Schaack and Larson* [1985] and *Rege and Sengupta* [1985]. The $M/M/\{N_i\}\otimes\{\underline{S}\}$ model integrates both these features, i.e., it keeps servers in reserve for emergencies, and it allows for multiple servers to be assigned to a single job.

The $M/M/\{N_i\}\otimes\{\underline{S}\}$ model must be considered an extension of a number of classical queueing models found in the literature. **Table 2.1** summarizes the most important of these special cases.

The two papers most relevant to this study are *Green* [1984] and *Schaack and Larson* [1985]. The $M/M/\{N_i\}\otimes\{\underline{S}\}$ model merges the simple cutoff model, $M/M/\{N_i\}$

# of priorities	$\mathbb{S}$	Cutoffs ?	Reference
1	$\sigma_{i1}=1, \forall i$	no	– M/M/m – *Erlang* [1917]
T	$\sigma_{i1}=1, \forall i$	no	*Cobham* [1954]
2	$\sigma_{i1}=1, \forall i$	yes	*Benn* [1966], *Jaiswal* [1971] *Descloux* in *Cooper* [1972/81] *Taylor & Templeton* [1980]
T	$\sigma_{i1}=1, \forall i$	yes	– M/M/$\{N_i\}$ – *Schaack & Larson* [1985] *Rege & Sengupta* [1985]
1	general	no	*Green* [1980]
T	general	no	*Green* [1984]
T	general	yes	– M/M/$\{N_i\}\otimes\{\underline{S}\}$ –

Table 2.1 – References

discussed in *Schaack and Larson* [1985], and the random-number-of-servers model proposed in *Green* [1984]. The former tackles the T-priority case with cutoffs where each arrival requires but a single server, while the latter develops results for a T-priority environment with stochastic number-of-servers requirements but no cutoffs. (To solve for the steady state probabilities and the waiting time distributions of the systems considered, one follows solution approaches based on M/G/1 queueing theory. This M/G/1 methodology shows promise in tackling other complicated Markovian queueing systems.)

We would like to draw the readers attention to a small difference in assumptions between our basic M/M/$\{N_i\}\otimes\{\underline{S}\}$ model (as described above) and the model described in *Green* [1984] (apart from the cutoff issue which is not addressed in the latter paper). Green considers a priority i call irrevocably "assigned" the moment all higher priority queues are empty, and one server is "free" (i.e. fewer than N_i servers are busy). If a higher priority call arrives while the priority i call is assigned, but not yet served (i.e., not all requested servers are available yet), Green queues the high-priority arrival. Our model in a sense allows preemption of low-priority calls that are assigned but not yet served, i.e., if a higher priority call arrives while a priority i call is assigned, but not yet served, we serve the higher priority arrival first: we deassign the priority i customer. (However, neither Green nor we allow preemption once "service" has actually started. The rationale behind this is that it is usually impractical or infeasible to recall police patrol units once they are active on the scene of an incident; remember, this is the reason why dispatchers would want to use cutoffs in the first place.) *Schaack* [1985] discusses in detail a family of queueing models that are extensions and variants of the basic $M/M/\{N_i\}\otimes\{\underline{S}\}$ model described here; included in this family is the direct extension of *Green* [1984], where assignment may not be preempted.

The $M/M/\{N_i\}\otimes\{\underline{S}\}$ model is akin to both bulk arrival and bulk service models, although it does not fit the standard mold of either of these models. Typically, in bulk arrival models, one arrival brings a (random) number of customers to the system; these customers usually get serviced independently by individual servers. In classic bulk service models, the server(s) service customers when a group of a certain size is waiting in queue. In the $M/M/\{N_i\}\otimes\{\underline{S}\}$ system, the arrival of a customer requesting k servers can be interpreted as the arrival of k quasi-customers requesting a single server. In that sense, $M/M/\{N_i\}\otimes\{\underline{S}\}$ is a bulk arrival model. These quasi-customers do not, however, start service independently of each other, as in classic bulk arrival models. Service starts simultaneously on all k quasi-customers. In that sense, $M/M/\{N_i\}\otimes\{\underline{S}\}$ is a bulk service model. It departs in two ways from the classic bulk service model: servers terminate service independently of each other, and, more significantly perhaps, the servers cannot select the group to be served by simply looking at the queue size. Thus while $M/M/\{N_i\}\otimes\{\underline{S}\}$ has features of both bulk arrival and bulk service models, it does not fit into the classic frame of either of these models. It is a hybrid, and interpretation in terms of bulk arrival or bulk service must be carefully worded. The reader may want to think of it as a bulk service model, in which the size of the group to be served depends on the type of the customers in queue (i.e., on the arrival process); all servers must begin service simultaneously on the group in question, one server to a quasi-customer, and servers terminate service on their quasi-customers independently (with identical exponential service time distributions).

3 Analysis of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ Model

This section is devoted to the mathematical analysis of the queueing model $M/M/\{N_i\}\otimes\{\underline{S}\}$, that addresses both the issues of efficiently implementing a preferential response policy (→cutoffs) and of assigning multiple response units when such an allocation scheme is deemed necessary (→random-number-of-servers requirements). The modeling issues were discussed in detail in Section 2.

We briefly recall the assumptions of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ model:

- N identical servers.
- T priority levels of customers.
  λ_i ≡ Poisson arrival rate of type i customers, $i = 1, 2, ..., T$.
  σ_{ik} ≡ probability that a priority i customer requires k servers.
  μ ≡ exponential service rate (identical for all priority levels and servers).
- Type i customers requiring k servers enter service immediately upon arrival only if fewer than N_i-k+1 servers are busy (where $0<N_T\leqq N_{T-1}\leqq ...\leqq N_2\leqq N_1=N$) and no calls of priority i or higher are backlogged; otherwise they join an infinite capacity queue of other priority i customers. The next of these customers to enter service, assuming she requests k servers, leaves the queue for the service facility at instants of server free-up arising when precisely N_i-k+1 servers are busy and all higher priority queues are empty (the service discipline is HOL by priority).
- Within a priority, the service order, unless specified otherwise, is assumed to be FCFS. Other disciplines, that are tractable for M/G/1 queues with exceptional first service in a busy period, are possible.

The $M/M/\{N_i\}\otimes\{\underline{S}\}$ model.

We have, in this basic version of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ model, assumed that the queue capacity is infinite. The model is similarly tractable for zero-capacity ("loss") systems, as illustrated in Section 3.6.

The model implicitly assumes that all servers servicing a particular customer finish service independently of each other. This may or may not be a reasonable assumption depending on the application, as we briefly discussed with respect to police patrol dispatching (in Section 2.1).

With the above assumptions, there is no permanent assignment of servers to a priority i customer requesting k servers until all k servers are "available" (i.e., until fewer than $N_i - k + 1$ servers become busy and the customer in question is the highest priority customer in line waiting to be served), at which time service begins. A low-priority customer that has been assigned a server but has not started service yet (i.e., not all servers have been assigned) will be preempted by any higher priority arrival. Under no circumstance, however, will preemption occur once actual service has started.

As an alternative to this assumption of preemptive assignment, one could consider a queueing policy that considers a customer irrevocably assigned upon the moment that one server becomes "available" (cf., e.g., *Green* [1984]). Under such a policy, upon the instant that a customer has been assigned one (out of k requested) server, she rates a higher priority than any customer that may enter the system subsequently. She is therefore given, upon assignment, access to all N servers in the system, not just to the cutoff number N_i corresponding to her original priority clearance. Until she has received her quota of servers (i.e., until she actually starts service), all other (arriving) calls must wait, regardless of their priority. In some sense, this latter policy forbids preemption on assignment, while the former (our default policy in this chapter) expressly allows it. The policy of nonpreemptive assignment and hybrid policies including features of both the preemptive and nonpreemptive assignment policies are mentioned in Section 5, but the reader is referred to *Schaack* [1985] for a detailed discussion of these alternative models.

3.1 The M/G/1 Approach

Our analysis of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ queueing system is based on the same M/G/1 approach that led to the successful solution of the simpler $M/M/\{N_i\}$ system *Schaack and Larson* [1985]. Albeit conceptually similar, the arguments that lead to the solution of the model with stochastic server requirements are substantially more delicate and involved. The $M/M/\{N_i\}$ system is skipfree positive as well as skipfree negative: To go from a state with k servers busy to a state with n servers busy ($n \neq k$), the system has to pass through a state with $n+1$ servers busy if $k > n$ (skipfree negative), or through a state with $n-1$ servers busy if $k < n$ (skipfree positive). The $M/M/\{N_i\}\otimes\{\underline{S}\}$ system lacks part of this property: it is not skipfree positive. Downward transitions are still skipfree in this model, but, unless $\sigma_{i1}=1$ for all $i \in \{1, 2, ..., T\}$, upward transition are not any more. However, enough structure is preserved to permit an analytical solution of the model along similar lines.

For the analytical developments of the following sections, it is helpful to view the M/M/$\{N_i\} \otimes \{\underline{S}\}$ queueing system in the same way we viewed the M/M/$\{N_i\}$ system: Customers of priority i are waiting in queue i and have no information about the queues of other priorities, which form in other waiting rooms of the service facility (**Figure 3.0**). While they are in their waiting room, they firmly

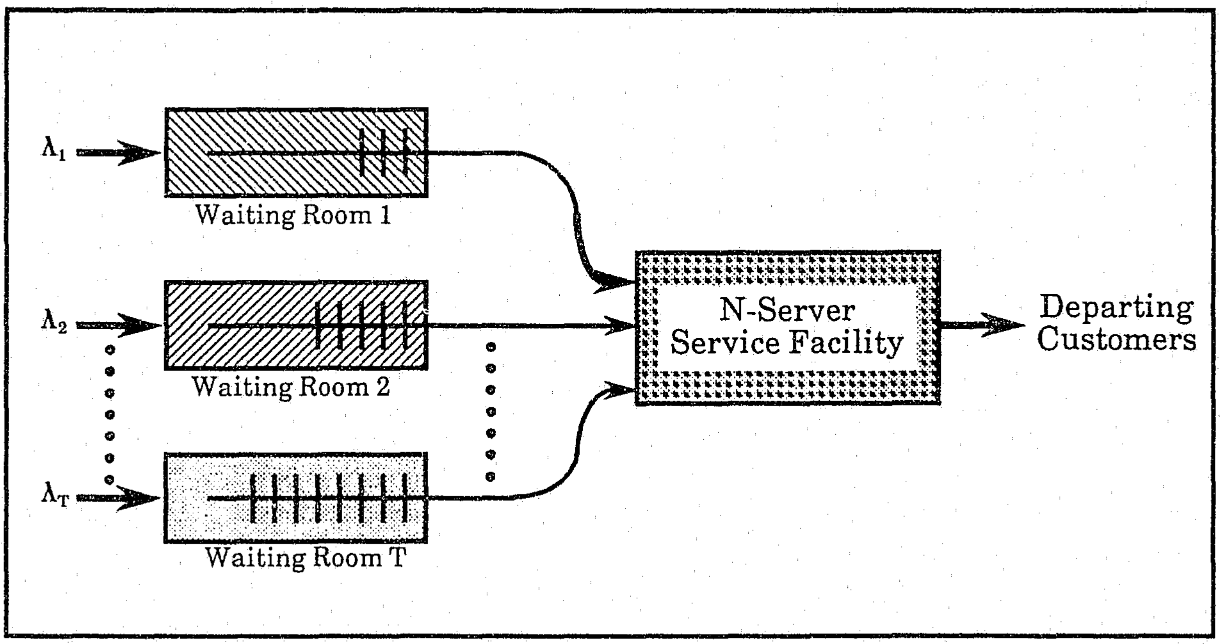


Figure 3.0

Arrival Streams of Different Priorities Enter Separate Queues

believe that they wait in the only queue in the system. Therefore assume that the customers in queue i can only observe how their own queue behaves, i.e., when the next customer in their queue begins service. A priority i customer that arrives to a non-empty queue observes that the times between successive "move-ups" in queue position (say from position k to $k-1$, $k \geqq 1$) are independent, identically distributed (i.i.d.) with a general distribution for the time between move ups. This queueing behavior is similar to that of an M/G/1 system (except in general for the first customer who incurs a delay in a busy period, as we shall see shortly), where G depicts a general service time distribution represented here by the time between successive move-ups. G is not, however, the distribution of time actually spent in service by a type i customer. The observed G for times between move-ups is in fact the probability distribution of a delay cycle sustained by higher priority arrivals (i.e., of priorities $1, 2, ..., i-1$), whose existence our priority i customer is unaware of. We shall formalize these concepts as we go along.

In Section 3.2 our aim is to determine the distribution of a family of delay cycles of importance to a priority i customer. These delay cycles are used in Section 3.3 to determine the probability that a random (tagged) customer arrives at the queueing system while

(i) the system is congested (for the customer's priority clearance), or,

(ii) the system is not congested **and** a certain number of servers are busy.

We shall shortly define, in more rigorous terms, what we understand by a congested system. The particular state description outlined above reflects the minimum amount of detail needed to account for the stochastic server requirements ($\underline{S}$). The probabilities computed in Section 3.3 are then used (in Section 3.4) for (un-)conditioning purposes, when we derive, again using our delay cycles, the waiting time distribution of a priority i customer.

In summary, the three analytical steps: (1) derivation of the queue move-up times, (2) computation of the steady state (time average; i.e., Poisson incidence) probabilities and (3) derivation of the waiting times in transform domain are essentially the same steps undertaken for the $M/M/\{N_i\}$ model, described in *Schaack and Larson* [1985]. All steps are complicated by the fact that the upward transitions are not skipfree positive. In Step (1), the recursions defining the queue move-up times become more involved. In Step (2), our state description must reflect more detail than it did for $M/M/\{N_i\}$. The argumentation used for the simpler model is ineffective in the more convoluted setting of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ model. Finally, in Step (3), we must recognize that the first "virtual service" time (read: queue move-up time) in a "busy period"* is, in general, different from the remaining "service" times, and that appropriate adjustments to the M/G/1 results must be made. M/G/1 queues with exceptional first service have been studied in the literature (e.g., *Welch* [1964]), and waiting time results for the $M/M/\{N_i\}\otimes\{\underline{S}\}$ model can be derived by analogy with these models.

* The term "busy period" is used rather loosely here. Appropriate concepts are defined rigorously in the next section.

3.2 Elementary Delay Cycles

3.2.1 Definitions

Definition 3.1: Unless stated otherwise, we define **service completion** instants to be time instants at which some server finishes servicing some customer.

Indeed, in the M/M/{N_i}⊗{S} queueing system, service completions are well defined in terms of servers, but not in terms of customers. In terms of customers one would need to specify whether one means the instant at which the first, ..., or the last (of k) servers servicing a given customer finishes his job.

We shall make extensive use of the following default convention for summations and products: Whenever the lower bound on a summation (respectively, a product) exceeds the upper bound, the value of the summation (respectively, the product) is taken to be zero (respectively, one):

$$\sum_{i=b}^{a} x_i \equiv 0 \quad \text{and} \quad \prod_{i=b}^{a} x_i \equiv 1 \quad \text{if } b>a .$$

We also, by convention, denote the Laplace-Stieltjes transform of the distribution of a random variable X by $X^*(s)$.

Table A.1 in the appendix summarizes the plethora of variable definitions that we introduce throughout this paper. The reader will probably find it convenient to turn to this table as an aide-mémoire.

In this section we shall endeavour to obtain the probability distributions of certain *elementary delay cycles** that will be useful in analyzing the M/M/{N_i}⊗{S} queueing system. These elementary delay cycles are essentially building blocks for the following two sections on steady state probabilities and waiting time distributions.

* For an introduction to standard delay cycles, the unfamiliar reader is referred to *Kleinrock* [1975], Vol. 2, pp. 111ff.

Definition 3.1: Elementary delay cycles $R_{i,n}$

Assume that all arrival streams of priorities $i+1$ through T are suppressed from the system after time t. Let $(n; q_1, q_2, q_3, ..., q_i)$ denote a (micro-)state in this system, where n is the number of busy servers, and q_j is the number of customers of priority j in queue, for $j \in \{1, 2, ..., i\}$. Suppose at time t, all queues (of priority 1 through i) are empty, and there are n servers busy, i.e., the system is in state $(n; 0, 0, 0, ..., 0)$. Let $(n; q_1, q_2, ..., q_j, \bullet, \bullet, ..., \bullet)$ denote the subspace "n servers busy, q_k customers in queue, for $k \in \{1, 2, ..., j\}$, any number of customers in queue for $k \in \{j+1, ..., i\}$".

Let r denote the lowest priority whose cutoff N_r is at least equal to n, i.e., $r \equiv \max\{j, N_j \geq n\}$. Let ${}_rX_{in}$ denote the first passage time from state $(n; 0, 0, 0, ..., 0)$ to state $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$, i.e., to absorption in the subspace $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$.

Let $a_{r+1}{}^c$ denote the number of arrivals of priority $r+1$ during ${}_rX_{in}$. Let ${}_{r+1}X_{in}$ denote the first passage time from state $(N_{r+1}; 0, ..., 0, q_{r+1} = a_{r+1}{}^c, \bullet, ..., \bullet)$ to state $(N_{r+1}; 0, ..., 0, q_{r+1} = 0, \bullet, ..., \bullet)$.

Similarly, for $k \in \{i, ..., r+1\}$, let $a_k{}^c$ denote the number of arrivals of priority k during ${}_rX_{in} + {}_{r+1}X_{in} + ... + {}_{k-1}X_{in}$.† Let ${}_kX_{in}$ denote the first passage time from state $(N_k; 0, ..., 0, q_k = a_k{}^c, \bullet, \bullet, ..., \bullet)$ to state $(N_k; 0, ..., 0, q_k = 0, \bullet, \bullet, ..., \bullet)$.

Then we define the *elementary delay cycle* $R_{i,n}$ by $R_{i,n} \equiv {}_rX_{in} + {}_{r+1}X_{in} + ... + {}_iX_{in}$.

This definition calls for a number of comments:

(1) Because the $M/M/\{N_i\} \otimes \{\underline{S}\}$ system is skipfree negative with respect to the number of busy servers, during one of the first passage times defined above, say from state $(n; 0, 0, 0, ..., 0)$ to state $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$, no state of the form $(m; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$, with $m < n-1$, can be reached before state $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$ is reached; i.e, before the end of the first passage time in question. The destination state $(n; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$ is

† This is an abuse of notation; to be rigorous, we should talk about the number of arrivals during the union of time intervals underlying the ${}_jX_{in}$'s.

always reached upon a service completion from some (micro-)state in the subspace $(n; 0, \ldots, 0, q_r = 0, \bullet, \bullet, \ldots, \bullet)$.

(2) $R_{i,n}$ can be interpreted as a delay cycle with initial delay the time until the first service completion from state $(n; 0, 0, 0, \ldots, 0)$, and with a delay busy period sustained by Poisson arrivals of priorities 1 through i. We call these delay cycles *elementary* because their initial delay is simple; and because, as we shall see shortly, they are elementary building blocks for more complicated first passage times.

(3) Notice that $R_{i,n}$ is typically not one continuous interval of time, but a union of time intervals separated by other time intervals, all of which belong to the same renewal cycle (where we define a renewal cycle as bounded by entries int state $(n = 0; 0, \ldots, 0)$). This feature is illustrated in **Example 3.1**.

For computational purposes it is useful to extend the definition of elementary delay cycles to include the following ("priority 0") boundary condition:

Definition 3.2: (Elementary Delay Cycles) $\mathbf{R_{0,n}}$

Suppose that, at time t, there are n servers busy. Then $\mathbf{R_{0,n}}$ is defined as the **time until the first service completion** subsequent to t, for $0 < n \leq N_1 = N$.

$R_{0,n}$ can be viewed in the following way, consistent with our definition of elementary delay cycles:

Assume that all arrival streams are suppressed from the system after time t. Let (n) denote a (micro-)state in this system, where n is the number of busy servers. Then $R_{0,n}$ is the first passage time from state (n) to state $(n-1)$ in this system.

$R_{0,n}$ can also be viewed as a delay cycle with initial delay the duration until the first service completion, and with delay busy period sustained by arrival streams of priority higher than priority 1 (i.e., of rate zero: not sustained at all).

Example 3.1

Suppose $T=2$, $N=3$, $N_2=1$, $n=N$, $i=2$. At time t the process is in state "3 servers busy, nobody in queue". Let us look at one particular occurrence of this process: At time t', the first service completion occurs (i.e., the initial delay is equal to $t'-t$). A single (tagged) arrival, of priority 2 and requesting a single server, arrives in the interval [t,t'], and no further arrival occurs for a very long time after t'. At time t', the system contains work due to the one tagged priority 2 arrival. However, because $N_2 = 1 < 3$, this work cannot be resorbed until some later date t" when a service completion occurs from state "1 server busy". Because (in the present occurrence) of the process no further arrival is due for a long time, the following service completion, from state "1 server busy" to "0 servers busy", at time t'", marks the end of the time period ${}_1X_{23}$. Regardless of the number of arrivals subsequent to t" (zero in this occurrence of the process), t' and t" are at the very least (in this system) separated by two service completions, one from from state "3 servers busy" to state "2 servers busy" and one from state "2 servers busy" to state "1 server busy"; therefore t" > t'. t" marks the beginning of the first time period ${}_2X_{23}$,when the system takes care of the work added by the occurrence of our tagged arrival, $R_{2,3}$ ($R_{2,N}$) is the reunion of two non-contiguous time periods, [t,t'] and [t",t'"] as illustrated by **Figure 3.1**. (For other occurrences of the process, $R_{2,N}$ may be reduced to [t,t'].)

The definition of the elementary delay cycles implies the following result:

<u>Result 3.1</u>: **In a system with arrival streams of priorities i + 1 through T suppressed, the first passage time, $FPT_{i,n,m}$, from state "n servers busy and all queues (of priority 1 through i) empty" to state "m servers busy and all queues (of priority 1 through i) empty" is given by**

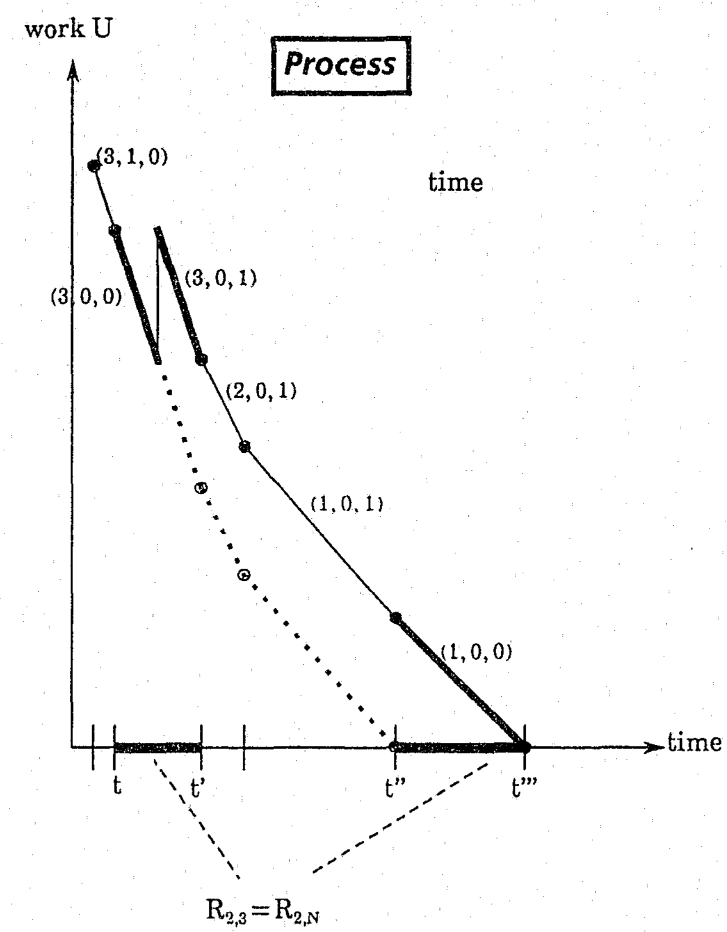
$$FPT_{i,n,m} = \sum_{l=m+1}^{n} R_{i,l} \quad \text{for } m < N_i \leq n.$$

We argue this by induction on n:

Arrival streams of priority i + 1 through T are non-existent.

First, assume, $n = m + 1$. $R_{i,n} \equiv {}_rX_{in} + {}_{r+1}X_{in} + ... + {}_iX_{in}$; but $r \equiv \max\{j | N_j \geq n\} = i$, thus $R_{i,n} \equiv {}_rX_{in} = {}_iX_{in}$, and $R_{i,n}$ is the first passage time from state $(n; 0, 0, 0, ..., q_i = 0)$ to $(m; 0, 0, 0, ..., q_i = 0)$.

Now, suppose the result is true for $n = s - 1$. Let us prove that it holds for $n = s$. Let $r \equiv \max\{j | N_j \geq s\}$. Let ${}_rX_{is}$ be the first passage time from $(s; 0, 0, 0, ..., q_i = 0)$ to $(s - 1; 0, ..., q_r = 0, \bullet, ..., \bullet)$. Let $a_{r+1}{}^c$ denote the number of arrivals of priority r during ${}_rX_{is}$. Let us put all these arrivals in a dark room and forget about them temporarily; i.e., temporarily, it is as if the system were in state $(n - 1; 0, ..., q_i = 0)$. The first passage time from this state to $(s; 0, 0, 0, ..., q_i = 0)$ is just $FPT_{i,s-1,m}$; by our induction hypothesis, this is also $R_{i,s-1} + ... + R_{i,m+1}$. Now,



Notation

$(n,q_1,q_2) \equiv$ (# of busy servers, # of priority 1 customers in queue, # of priority 2 cust. in queue)

Figure 3.1 – Illustration of $R_{i,n}$
(Data from Example 3.1 – Two priority system: $T=2$; $1=N_2<N=3$)

let us look at the contribution of our locked-up priority $r+1$ customers. They add a time interval of length ${}_{r+1}X_{is}$ to $FPT_{i,s-1,m}$, where ${}_{r+1}X_{is}$ is distributed as a first passage time from state $(N_{r+1}; 0, ..., 0, q_{r+1}=a_{r+1}^c, \bullet, ..., \bullet)$ to state $(N_{r+1}; 0, ..., 0, q_{r+1}=0, \bullet, ..., \bullet)$. Similarly, for $k \in \{r+2, ..., i\}$, let a_k^c denote the number of arrivals of priority k during ${}_rX_{is} + {}_{r+1}X_{is} + ... + {}_{k-1}X_{is}$ (or, rather, the underlying union of intervals), and assume they are all locked up in a room until the servers can turn their attention to them. Then the length of time added by the need to service the a_k^c priority k customers is given by ${}_kX_{is}$, which is distributed as a first passage time from state $(N_k; 0, ..., 0, q_k=a_k^c, \bullet, \bullet, ..., \bullet)$ to state

$(N_k; 0, ..., 0, q_k = 0, \bullet, \bullet, ..., \bullet)$. Thus

$$FPT_{i,s,m} = FPT_{i,s-1,m} + \sum_{k=i}^{r} {}_kX_{i,l} = FPT_{i,s-1,m} + R_{i,s} \ ,$$

or, by our induction hypothesis,

$$FPT_{i,s,m} = \sum_{l=m+1}^{s} R_{i,l} \ ,$$

which concludes the induction. ∴

This result is the key to the success of the first passage times method that we use for the cutoff problems. The *elementary delay cycles* $R_{i,n}$ are of interest only in as far as they allow us to successfully evaluate the actual downward first passage times $FPT_{i,n,m}$. These are crucial random variables for the derivation of both the steady state probabilities (Section 3.3) and the waiting time distributions (Section 3.4). While the $FPT_{i,n,m}$'s are easier to understand, the $R_{i,n}$'s are easier to evaluate. Moreover, as one can observe in Sections 3.2.2 and 3.3, there are $O(N^2)$ $FPT_{i,n,m}$'s that need to be evaluated, but only $O(N)$ $R_{i,n}$'s, which, all other considerations aside, offers an added incentive to use the (perhaps less intuitive) elementary delay cycles, $R_{i,n}$, rather than the clumsier multi-stage first passage times, $FPT_{i,n,m}$. In order not to further add to an already complex notation, we will henceforth reason in terms of elementary delay cycles, $R_{i,n}$, and dispense with the $FPT_{i,n,m}$ notation.

We finally introduce one last definition that will be useful in the next section:

Definition 3.3: We define the random variable $V_{i,n}$ as the **time until the next transition** from state "n servers busy" *in a system with arrival streams of priority 1 through i only*, and this for $i \in \{1, 2, ..., T\}$ and $n \in \{1, 2, ..., N_i - 1\}$.

3.2.2 Derivation of the Elementary Delay Cycles $R_{i,n}$

We briefly recall, in **Table 3.1**, the definitions of the most important random variables defined in the previous section:

$\mathbf{R_{i,n}}$	$\equiv$	*elementary delay cycle* from state "n servers busy, nobody in queue", in a system with no arrival streams of priority lower than i, for $i\in\{0, 1, 2, ..., T\}$ and $n\in\{1, 2, ..., N\}$.
$\mathbf{V_{i,n}}$	$\equiv$	time until next transition from state "n servers busy" in a system with no arrival streams of priority lower than i, for $i\in\{1, 2, ..., T\}$ and $n\in\{1, 2, ..., N_i - 1\}$.
$\mathbf{X^*(s)}$	$\equiv$	Laplace-Stieltjes transform of the distribution of a random variable X.

Table 3.1 – A few Definitions

The Elementary Delay Cycles $R_{0,N}$

From the definition of $R_{0,n}$, we trivially obtain the Laplace-Stileltjes transform:

$$R^*_{0,n}(s) = \frac{n\mu}{n\mu + s} \quad for\, n = 1, 2, ..., N. \tag{3.1}$$

The Elementary Delay Cycles $R_{1,N}$

We now derive the Laplace-Stieltjes transform of $R_{1,n}$, the *elementary* delay cycle from state "n servers busy and no priority 1 customers queued" (to state "n − 1 servers busy and no priority 1 customers queued"), for a system with arrivals of priority 2 through T suppressed.

First consider $R_{1,N}$. Let there be N servers busy and no customers in queue at time t. Let X_1 be the duration of time until the first service completion. Let K_j be the number of customers of priority i requesting j servers that arrived during X_1, and let K denote the total number of customers of priority 1 that arrived during X_1 ($K \equiv K_1 + ... + K_N$). As they arrived, these K customers were conveniently locked up in a big dark room.

After X_1 has ended, we retrieve one by one the K arrivals from the dark room (in any order: e.g., LCFS) upon time instants at which the system enters a state where

(i) N servers are busy,

and

(ii) no customers, except others in the dark room, are waiting to be served.

Let us focus on a particular customer as it is retrieved. We start a clock upon the moment of retrieval. If the retrieved customer requests j servers, she must wait until sufficient servers become available. However, not only must she wait until sufficient servers are available, we also decide that if there are any more arrivals, they (and she) are served in a LCFS manner. In other words, she will enter service only upon the first time instant when

(i) j servers are available,

and

(ii) no customers, except in the dark room, are waiting to be served.

In effect, this LCFS policy ensures that all "descendents" of this customer (i.e., all new arrivals that arrive while she is retrieved and waits for service) enter service before her, in a LCFS manner. We stop the clock when our retrieved customer finally enters service.

In order to compute the elapsed time, consider that at each stage during which the system tries to free another server for our retrieved customer, that is during inter-service-completion intervals, more customers may arrive. Because, under our modified (work conserving!!!) policy, they get served in a LCFS manner, they in a sense "preempt" the retrieved customer "on assignment". These new arrivals consequently contribute a delay cycle at each of the j stages. The delay cycle at stage k is distributed as $R_{1,k}$, by definition of $R_{1,k}$. Therefore, the added work contributed to the current renewal cycle by our retrieved customer contributes to $R_{1,N}$ a random length of time distributed as

$$R_{1,N}+R_{1,N-1}+\ldots+R_{1,N-j+1}\ .$$

The distribution of this sum of independent random variable, in transform domain, is given by:

$$R^*_{1,N}(s)R^*_{1,N-1}(s)\cdot\ \cdot\ \cdot\ R^*_{1,N-j+1}(s)=\prod_{n=N-j+1}^{N}R^*_{1,n}(s)$$

As all other customers in the dark room are, in turn, retrieved, they add similar time intervals to $R_{1,N}$. Therefore, remembering to count the duration X_1 of the time until the first service completion (from "N servers busy") that started all this, we can write, for $R_{1,N}$ conditioned upon K_j and X_1:

$$E\left[e^{-sR_{1,N}}\ \middle|\ X_1=y,(K_j=k_j\,,\forall j\in\{1,\ldots,N\})\right]=e^{-sy}\prod_{j=1}^{N}\left[\prod_{n=N-j+1}^{N}R^*_{1,n}(s)\right]^{k_j}$$

Unconditionining on the K_j's, we obtain:

$$E\left[e^{-sR_{1,N}} \,\middle|\, X_1 = y\right] = e^{-sy} \prod_{j=1}^{N} \left[\sum_{k_j=0}^{\infty} e^{-\lambda_1 \sigma_{1j} y} \frac{(\lambda_1 \sigma_{1j} y)^{k_j}}{k_j!} \left[\prod_{n=N-j+1}^{N} R^*_{1,n}(s) \right]^{k_j} \right]$$

or,

$$E\left[e^{-sR_{1,N}} \,\middle|\, X_1 = y\right] = e^{-sy} \prod_{j=1}^{N} \left[e^{-y\left(\lambda_1 \sigma_{1j} - \lambda_1 \sigma_{1j} \prod_{n=N-j+1}^{N} R^*_{1,n}(s)\right)} \right]$$

or,

$$E\left[e^{-sR_{1,N}} \,\middle|\, X_1 = y\right] = e^{-y\left(s + \lambda_1 - \lambda_1 \sum_{j=1}^{N} \sigma_{1j} \prod_{n=N-j+1}^{N} R^*_{1,n}(s)\right)}$$

Then, unconditioning on X_1 (i.e., $R_{0,N}$), we find an equation defining the transform of the distribution of $R_{1,N}$:

$$R^*_{1,N}(s) = E\left[e^{-sR_{1,N}}\right] = R^*_{0,N}\left(s + \lambda_1 - \lambda_1 \sum_{j=1}^{N} \sigma_{1j} \prod_{n=N-j+1}^{N} R^*_{1,n}(s)\right) \tag{3.2}$$

A different (simpler) argument is used to derive $R_{1,N-1}$. Because $R_{1,N-1}$ is an actual first passage time, the argument is not quite so tricky. In the system where all arrivals of priority lower than priority 1 are suppressed, we condition $R_{1,N-1}$ on the nature of the transition out of state "N − 1 servers busy", i.e., whether it is a service completion (a downward transition) or an arrival (an upward transition). To simplify our equations, we introduce the Laplace-Stieltjes transform of the distribution of $V_{i,n}$. Because of the Markovian nature of the process, the transform is clearly given by

$$V^*_{i,n}(s) = \frac{\lambda^c_i + n\mu}{\lambda^c_i + n\mu + s} \quad where\, \lambda^c_i = \sum_{k=1}^{i} \lambda_k \quad .$$

Assume now that a (tagged) priority 1 customer requesting j servers arrives before one of the N − 1 busy servers can finish service. This arrival will not start service until the moment when exactly j servers would become available. Now assume that during the time our tagged customer has to wait for this moment, there arrive more priority 1 customers. Since the distribution of $R_{1,N-1}$ is independent of the order in which priority 1 customers are processed, let's process them in a LCFS manner. This results, conditionally on a first arrival requesting j servers, in a (LCFS) waiting time distribution (for our tagged customer) whose Laplace-Stieltjes transform is given by:

$$\prod_{n=N-j+1}^{N-1} R^*_{1,n}(s)\ .$$

Therefore, conditioning on the type of transition, we find that $R_{1,N-1}$ is determined by:

$$R^*_{1,N-1}(s)=\frac{(N-1)\mu}{\lambda_1+(N-1)\mu}V^*_{1,N-1}(s)+\frac{\lambda_1}{\lambda_1+(N-1)\mu}\sum_{j=1}^{N}\sigma_{1j}V^*_{1,N-1}(s)R^*_{1,N}(s)R^*_{1,N-1}(s)\prod_{n=N-j+1}^{N-1}R^*_{1,n}(s)$$

or,

$$R^*_{1,N-1}(s)=\frac{V^*_{1,N-1}(s)}{\lambda_1+(N-1)\mu}\left((N-1)\mu+\lambda_1\sum_{j=1}^{N}\sigma_{1j}R^*_{1,N}(s)R^*_{1,N-1}(s)\prod_{n=N-j+1}^{N-1}R^*_{1,n}(s)\right)\ . \tag{3.3a}$$

Similarly, for $R_{1,n}$ $(0<n<N-1)$, one can write (using the default convention that "empty" products are equal to 1):

$$R^*_{1,n}(s)=\frac{V^*_{1,n}(s)}{\lambda_1+n\mu}\left[n\mu+\lambda_1\sum_{j=1}^{N}\sigma_{1j}\left(\prod_{m=N-j+1}^{n}R^*_{1,m}(s)\right)\left(\prod_{m=n}^{min(N,n+j)}R^*_{1,m}(s)\right)\right] \tag{3.3b}$$

Equations (3.2) and (3.3), repeated below, completely define the generalized first passage times $R_{i,n}$, for $i=1$ and $0<n\le N$.

$$R^*_{1,N}(s)=E\left[e^{-sR_{1,N}}\right] = R^*_{0,N}\left(s+\lambda_1-\lambda_1\sum_{j=1}^{N}\sigma_{1j}\prod_{n=N-j+1}^{N}R^*_{1,n}(s)\right) \tag{3.2}$$

$$R^*_{1,n}(s)=\frac{V^*_{1,n}(s)}{\lambda_1+n\mu}\left[n\mu+\lambda_1\sum_{j=1}^{N}\sigma_{1j}\left(\prod_{m=N-j+1}^{n}R^*_{1,m}(s)\right)\left(\prod_{m=n}^{min(N,n+j)}R^*_{1,m}(s)\right)\right]\quad \text{for } 0<n<N \tag{3.3}$$

These expressions look rather repulsive; however, differentiating them (once) with respect to s and setting s to zero yields a linear system of equations, the variables of which are the (first) moments of the first passage times $R_{1,n}$, as the equations below show.

$$E\left[R_{1,N}\right]=E\left[R_{0,N}\right]\left(1+\lambda_1\sum_{j=1}^{N}\sigma_{1j}\sum_{n=N-j+1}^{N}E\left[R_{1,n}\right]\right)=\frac{1}{N\mu}\left(1+\lambda_1\sum_{j=1}^{N}\sigma_{1j}\sum_{n=N-j+1}^{N}E\left[R_{1,n}\right]\right), \tag{3.4}$$

and for $0<n<N$,

$$E\left[R_{1,n}\right]=\frac{1}{\lambda_1+n\mu}+\frac{\lambda_1}{\lambda_1+n\mu}\sum_{j=1}^{N}\sigma_{1j}\left[\left(\sum_{m=N-j+1}^{n}E\left[R_{1,m}\right]\right)+\left(\sum_{m=n}^{min(N,n+j)}E\left[R_{1,m}\right]\right)\right]\ . \tag{3.5}$$

This differentiation procedure can, of course, be repeated to obtain linear systems for the higher moments of $R_{1,n}$ (in terms of the lower moments).

The Generalized Delay Cycles $R_{i,n}$ (for $i>1$)

We now consider that all arrival streams of priority $i+1$ or lower are suppressed.

As before, let us first focus our attention on the random variables $R_{i,n}$, for the case where $N_i \leq n \leq N$. In order to be able to use an argument that parallels the argument that led to equation (3.2), we establish the following decomposition result, which is deeply rooted in the HOL structure of the system:

Result 3.2: The elementary delay cycle $R_{i,n}$ can be considered as a delay cycle with initial delay $R_{i-1,n}$, sustained by arrivals of priority i during $R_{i-1,n}$ and by arrivals of priorities 1 through i that arrive in subsequent intervals..

By definition,
$R_{i,n} = {}_rX_{in} + {}_{r+1}X_{in} + \ldots + {}_iX_{in}$, and $R_{i-1,n} = {}_rX_{i-1,n} + {}_{r+1}X_{i-1,n} + \ldots + {}_{i-1}X_{i-1,n}$. Notice that, by definition of the ${}_kX_{jn}$'s: ${}_kX_{in} = {}_kX_{i-1,n}$ for $r \leq k < i$. Thus, $R_{i,n} = R_{i-1,n} + {}_iX_{in}$. But ${}_iX_{in}$ is the first passage time from $(N_i, 0, 0, \ldots, q_i = a_i^c)$ to $(N_i - 1, 0, 0, \ldots, q_i = 0)$, where a_i^c is the number of arrivals of priority i during ${}_rX_{in} + {}_{r+1}X_{in} + \ldots + {}_iX_{i-1,n}$, i.e., during $R_{i-1,n}$. $\therefore$

This result now enables us to apply the same reasoning that we used previously on $R_{1,N}$ to the generalized first passage time $R_{i,n}$, **for $N_i \leq n \leq N$**. Paralleling the arguments that led to the derivation of equation (3.2), we can write:

$$R^*_{i,n}(s) = E\left[e^{-sR_{i,n}}\right] = R^*_{i-1,n}\left(s + \lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{m=N_i-j+1}^{N_i} R^*_{i,m}(s)\right) \qquad \text{for } N_i \leq n \leq N. \tag{3.6}$$

For $n \leq N_i - 1$, the derivation follows directly along the lines of the argument leading to equation (3.3), without envoking Result 3.2:

$$R^*_{i,n}(s) = \frac{V^*_{i,n}(s)}{\lambda_i^c + n\mu}\left[n\mu + \sum_{r=1}^{i} \lambda_r \sum_{j=1}^{N_r} \sigma_{rj}\left(\prod_{m=N_r-j+1}^{n} R^*_{i,m}(s)\right)\left(\prod_{m=n}^{\min(N_r, n+j)} R^*_{i,m}(s)\right)\right], \forall n \in \{1, \ldots, N_i - 1\} \tag{3.7}$$

As above, for $i=1$, differentiating these equation yields simple linear systems that are easily solved for the first (and higher) moments of the random variables $R_{i,n}$:

$$E\left[R_{i,n}\right] = E\left[R_{i-1,n}\right]\left(1+\lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{m=N_i-j+1}^{N_i} E\left[R_{i,m}\right]\right) \qquad \text{for } N_i \le n \le N, \tag{3.8}$$

and, for $n \in \{1, \ldots, N_i - 1\}$:

$$E\left[R_{i,n}\right] = \frac{1}{\lambda_i^c + n\mu}\left(1 + \sum_{r=1}^{i} \lambda_r \sum_{j=1}^{N_r} \sigma_{rj}\left[\left(\sum_{m=N_r-j+1}^{n} E\left[R_{i,m}\right]\right) + \left(\sum_{m=n}^{\min(N_r, n+j)} E\left[R_{i,m}\right]\right)\right]\right). \tag{3.9}$$

3.3 Steady State Incidence Probabilities

In this section we derive the steady state probabilities that a Poisson arrival sees the queueing system in a certain state. Indeed, in order to evaluate the distribution of the waiting time incurred by an arrival of a given priority, it is important to know with what probability this arrival finds the system "congested" for her priority class, and with what probability the system is "uncongested".

Definition 3.4: Congestion

The system is said to be congested for priority i if "at least one queue of priority i or higher is nonempty, **or** more than N_i servers are busy".

Similarly, we define a system state as uncongested for priority i if in that state "all queues of priority 1 through i are empty **and** at most N_i servers are busy".

Under our default service discipline (FCFS within a priority), if the system is congested (for priority i) when a (priority i) customer arrives to the system, the customer will have to wait. Indeed, either a customer of equal or higher priority is already in queue, or more than the cutoff N_i number of servers are busy. If on the other hand, the system is uncongested when a new arrival occurs, the arriving customer may or may not enter service immediately depending on the number of servers she requests. For example if an arrival requesting 2 servers occurs while the system is uncongested and $N_i - 3$ servers busy, this arrival can enter service right away, while if $N_i - 1$ servers were busy, she would have to wait. (It may appear counter-intuitive that the state "all queues of priority 1 through i are empty and N_i servers busy" is defined as uncongested for priority i, for any priority i arrival to that state will have to wait for service. For reasons of analytical tractability, it is preferable to include this state among the uncongested states.) Notice that the system is necessarily congested or uncongested for priority i at any point in time.

In order to simplify our argumentation somewhat, we introduce the following concepts and notations:

C_i	$\equiv$	(continuous) time period during which a system is congested for arriving priority i customers; by extension, C_i also denotes the macro-state "system congested for priority i".
U_i	$\equiv$	(continuous) time period during which the system is uncongested for priority i customers; by extension, U_i also denotes the macro-state "system uncongested for priority i".
p_i	$\equiv$	steady state probability that the system is in state C_i
q_i	$\equiv$	steady state probability that the system is in state U_i
$P_{n\|U_i}$	$\equiv$	probability that there are n servers busy, given that the system is uncongested for priority i, where $n \in \{0, 1, ..., N_i\}$.

Notice that the queueing system goes through cycles of congested and uncongested periods. In order to obtain the steady state probabilities p_i, q_i and $P_{n|U_i}$, we need only concern ourselves with a single $C_i \cup U_i$ cycle. The probabilities p_i and q_i are easily determined from

$$p_i = \frac{E[U_i]}{E[U_i]+E[C_i]} \qquad and \qquad q_i = 1 - p_i \ , \tag{3.10}$$

once we compute the expected values of the durations of blocked and uncongested periods for priority i. We shall now proceed to compute recursively, for all $i \in \{1, 2, ..., T\}$, $E[U_i]$, $E[C_i]$, p_i, q_i and $P_{n|U_i}$.

Outline of the derivations

- We have an initial lever on the steady state probabilities at the low priority end ($i = T$) of our state space (Section 3.3.1).
  - In steady state, the uncongested state, U_T, is always entered through state "N_T servers busy, all queues empty". It is easy to evaluate , using standard Markovian methods, how often the system visits state "n servers busy" while it is uncongested (U_T). The holding times per visit in these states are known (they are exponential with rates $\lambda_T^c + n\mu$). The expected holding times and the expected number of visits enable us to derive the steady state probabilities that n servers are busy, given that the system is uncongested ($P_{n|U_T}$) for priority T; and, similarly, they yield the expected sojourn time in state U_T ($E[U_T]$)

- The expected sojourn time in a congestion period, $E[C_T]$, is obtained by direct probabilistic arguments. Conditioning on what state in U_T the transition to C_T is initiated from, and using the *elementary delay cycles* derived in the previous section, one finds the expected duration of a congestion period ($E[C_T]$).
- Finally, the incidence probabilities p_T and q_T are easily determined from equation (3.10).

- We then proceed by induction from priority $i+1$ to priority i (Section 3.3.2).
  - We investigate the congestion period C_{i+1} by looking at substates of C_{i+1}. These substates are (i) C_i, the congestion state for priority i ($C_i|C_{i+1}$), and (ii) the uncongested state (U_i) with n busy servers, but congested for priority $i+1$ ($U_i\&n|C_{i+1}$). Using a conceptually similar, but substantially more involved Markovian approach than for $i=T$, we count the number of visits to these substates during one occupancy of C_{i+1}. The expected holding times in all but one of these states (state $C_i|C_{i+1}$) are known. The expected holding time in $C_i|C_{i+1}$ can be obtained from a conservation equation based on the expected duration of C_{i+1} (this is part of our induction hypothesis). Armed with these expected numbers of visits and holding times, it is then easy to obtain the probabilities that n servers are busy and the system is uncongested for priority i, given that the system is congested for priority $i+1$ ($U_i\&n|C_{i+1}$) and the probability that the system is congested for priority i, given that it is congested for priority $i+1$ ($C_i|C_{i+1}$).
  - Finally, unconditioning on incidence into U_{i+1} or $U_i\&n|C_{i+1}$, with $n \leq Ni$, one finds p_i, the steady state probability of an uncongested period for priority i. Further careful unconditioning yields the steady state probabilities that n servers are busy, given that the system is uncongested for priority i ($P_{n|U_i}$).
  - This concludes our induction argument. All quantities of interest can now be computed recursively.

Let us now turn to the derivations proper, starting with $i = T$, which case yields the boundary condition from which we start our recursive procedure. Let us concentrate on a priority T arrival, and on the system states that are of importance to her. It is convenient to assume that priority T customers can only tell how many servers are busy when the system is uncongested (i.e., states $n|U_T$, for $0 < n \leq N_T$). We therefore first focus on whether the system is or is not congested; next, if it is not congested, we focus on how many servers are busy.

3.3.1 The System "Seen" by Priority T Customers

- Let us first think about the **uncongested macro-state** U_T, specifically how it is entered, and how it is left. In steady state, U_T starts with a transition from C_T to state "N_T servers busy and all queues empty". While U_T lasts, no more than N_T servers are busy, and no queues (of any priority) form. As soon as a queue forms or more than N_T servers become busy the system enters a "congested period" C_T.

Now assume that the system (in steady state) is uncongested. What is the probability that there are exactly n servers busy? -This question can be answered easily if we know the expected number of visits to each of the states "0 servers busy" through "N_T servers busy" during one U_T period. Let $\underline{\mathbf{K}}$ be the (N_T+1)-by-(N_T+1) matrix defined by $K_{mn} \equiv$ Prob("system is still uncongested and n servers are busy after the next transition" *given* "system is now uncongested and m servers are busy"), for $(m,n) \in \{0, 1, ..., N_T\}^2$. The transition probability matrix $\underline{\mathbf{K}}$ is given by:

$$
\begin{array}{cccccc}
0 & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i1}}{\lambda_T^c} & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i2}}{\lambda_T^c} & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i3}}{\lambda_T^c} & \cdots & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i,N_T}}{\lambda_T^c} \\
\dfrac{\mu}{\lambda_T^c+\mu} & 0 & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i1}}{\lambda_T^c+\mu} & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i2}}{\lambda_T^c+\mu} & \cdots & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i,N_T-1}}{\lambda_T^c+\mu} \\
0 & \dfrac{2\mu}{\lambda_T^c+2\mu} & 0 & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i1}}{\lambda_T^c+2\mu} & \cdots & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i,N_T-2}}{\lambda_T^c+2\mu} \\
0 & 0 & \dfrac{3\mu}{\lambda_T^c+3\mu} & 0 & \cdots & \dfrac{\sum_{i=1}^{T} \lambda_i \sigma_{i,N_T-3}}{\lambda_T^c+3\mu} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & \dfrac{N_T\mu}{\lambda_T^c+N_T\mu} & 0
\end{array}
$$

$\underline{\mathbf{K}}$ is the transition probability matrix of a discrete trial Markov process with (artificial) trap state C_T, with the row and column corresponding to the congestion state, C_T, removed. Define the matrix $\underline{\mathbf{L}} \equiv (\underline{\mathbf{I}} - \underline{\mathbf{K}})^{-1}$. (Since $\underline{\mathbf{K}}$ is a substochastic matrix, the absolute values of its eigenvalues are strictly smaller than 1; therefore $\underline{\mathbf{I}} - \underline{\mathbf{K}}$ is invertible, which guarantees the existence of $\underline{\mathbf{L}}$.) For a Markov process with absorbing (or trapping) states, $\underline{\mathbf{L}}$ yields the expected number of visits to states 0 through N_T during one occupancy of macro-state U_T; more precisely, L_{mn} is the expected number of visits to state "n servers busy and system has not left U_T", given the system started in state "m servers busy and system in U_T".

The holding time in substate "n servers busy and system in U_T" is exponentially distributed with rate $\lambda_T^c + n\mu$. Since in steady state, the queueing system always enters macro-state U_T (from macro-state C_T) through substate "N_T servers busy", we can now write, for the expected duration of an uncongested period:

$$E[U_T] = \sum_{n=0}^{N_T} \frac{L_{N_T,n}}{\lambda_T^c + n\mu} \quad . \tag{3.11}$$

Therefore $P_{n|U_T}$, the steady state probability that there are n busy servers at a random time during an uncongested period U_T, is given by:

$$P_{n|U_T} = \frac{\dfrac{L_{N_T,n}}{\lambda_T^c + n\mu}}{\displaystyle\sum_{k=0}^{N_T} \frac{L_{N_T,k}}{\lambda_T^c + k\mu}} \qquad , \; n \in \{0, \ldots, N_T\} \tag{3.12}$$

- Now, let us focus on the **congested macro-state** C_T, and, more precisely, on the expected duration of a congested period, $E[C_T]$. In steady state, a congested period, C_T, begins when an arrival (of any priority) requesting more than $N_T - n$ servers occurs while the system is uncongested and n servers are busy, for $0 \leq n \leq N_T$. We refer to this arrival as the arrival that triggers the congested period.

Suppose C_T is triggered by a priority i arrival requesting k servers to state "n servers busy" in U_T. Because the process is skipfree negative, C_T will last until the system drops down to state "N_T servers busy" again, with all queues empty. Using the generalized delay cycles derived in Section 3.2, we can write for the Laplace-Stieltjes transform of the distribution of C_T, conditional upon the triggering event, as:

$$E\left[e^{-sC_T} \mid i, k, n\right] = \left(\prod_{m=N_i-k+1}^{n} R^*_{T,m}(s)\right)\left(\prod_{m=N_T+1}^{min(N_i, n+k)} R^*_{T,m}(s)\right) . \tag{3.13}$$

Indeed, the system must first try to free a sufficient number (N_i-k) of servers to start service on the triggering customer. As this customer enters service, the system must drop down to state N_T servers busy again to leave the congested period. In the meantime there may have arrived additional customers. These introduce delay busy periods that must finish before the system becomes uncongested again. **Figure 3.2** illustrates this process graphically, using the elementary delay cycles introduced in Section 3.2:

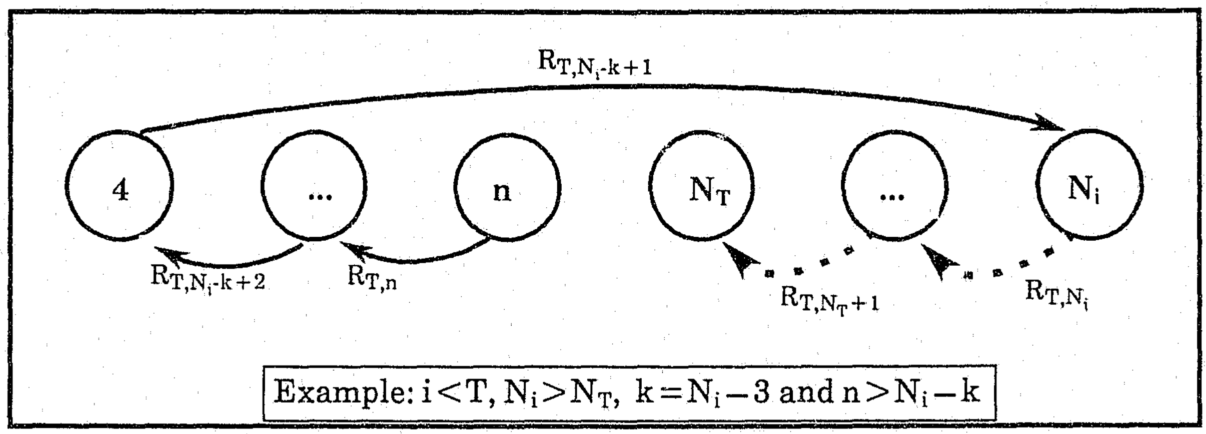


Figure 3.2 – C_T triggered by a priority i customer requesting k servers.

In order to uncondition on the triggering event, we need to know the probability that a congested period C_T is triggered by an arrival of priority i requesting k servers to substate "n servers busy" (in an uncongested period). This probability is simply given by:

$$\frac{P_{n|U_T}\lambda_i\sigma_{ik}}{\sum_{m=0}^{N_T} P_{m|U_T}\sum_{i=1}^{T}\lambda_i\sum_{j=N_T-m+1}^{N_i}\sigma_{ij}}\ , \quad \text{for } 0\le n\le N_T \text{ and } N_T-n+1\le k\le N_i\ .$$

Unconditioning on the properties of the Poisson arrival that triggers C_T we can write, using equation (3.13):

$$C_T^*(s)=\frac{\sum_{n=0}^{N_T} P_{n|U_T}\sum_{i=1}^{T}\lambda_i\sum_{k=N_T-n+1}^{N_i}\sigma_{ik}\left(\prod_{m=N_i-k+1}^{n} R_{T,m}^*(s)\right)\left(\prod_{m=N_T+1}^{\min(N_i,n+k)} R_{T,m}^*(s)\right)}{\sum_{m=0}^{N_T} P_{m|U_T}\sum_{i=1}^{T}\lambda_i\sum_{i=N_T-m+1}^{N_i}\sigma_{ij}} \qquad (3.14)$$

which easily yields, by differentiation, the expected value, $E[C_T]$, of the duration of a congestion period:

$$E[C_T] = \frac{\sum_{n=0}^{N_T} P_{n|U_T} \sum_{i=1}^{T} \lambda_i \sum_{k=N_T-n+1}^{N_i} \sigma_{ik}\left(\left(\sum_{m=N_i-k+1}^{n} E[R_{T,m}]\right)+\left(\sum_{m=N_T+1}^{min(N_i,n+k)} E[R_{T,m}]\right)\right)}{\sum_{m=0}^{N_T} P_{m|U_T} \sum_{i=1}^{T} \lambda_i \sum_{i=N_T-m+1}^{N_i} \sigma_{ij}} . \quad \textbf{(3.15)}$$

- Equations (3.10), (3.11)and (3.15) now enable us to compute the **probability** p_T (respectively, q_T) **that a random priority T arrival finds the system uncongested** (respectively, congested):

$$p_T = \frac{E[U_T]}{E[U_T]+E[C_T]} \qquad and \qquad q_T = 1 - p_T \ . \quad \textbf{(3.16)}$$

This completes the derivation of the steady state probabilities that we sought for priority T. Based on the boundary conditions for i=T (equations (3.11-12) and (3.15-16)), we proceed to derive, by induction, the same probabilities for customers of priorities 1 through T−1. (We actually work backwards: from T−1 to 1.) Suppose we know the following quantities for priority i+1: $E[C_{i+1}]$, $E[U_{i+1}]$, p_{i+1}, q_{i+1} and $P_{n|U_{i+1}}$., we now derive recursive relationships that define these same quantities ($E[C_i]$, $E[U_i]$, p_i, q_i and $P_{n|U_i}$) for priority i.

Again, assume a priority i arrival can see substructure (i.e., how many servers are busy) when the system is uncongested (for priority i), but not when it is congested. What happens within the macro-state C_i, is invisible to priority i customers.

3.3.2 The System "Seen" by Priority i Customers ($1 \leqq i < T$)

If the system is uncongested for priority $i+1$ (U_{i+1}), it is necessarily uncongested for priority i (U_i). If, however, the system is congested for priority $i+1$ (C_{i+1}), it may be either congested or uncongested for priority i, depending on the state of the queues and the number of busy servers. In order to gain more information on U_i and C_i, we therefore focus on state C_{i+1}.

Definition 3.5: For notational convenience, define, for this section (3.3.2), state "m" to be the state "**m servers busy and the system is in macro-state U_i, given that the system is in macro-state C_{i+1}**" for $1 \leq m \leqq N_i$.

And define "N_i+1" to be the state "**the system is in macro-state C_i, given that it is in macro-state C_{i+1}**".

We draw the reader's attention to the fact that although we write "m" for convenience, state "m" is conditioned on macro-state C_{i+1}. More importantly, we would like to stress that state "$\mathbf{N_i+1}$" does not, in general, refer to a state where N_{i+1} servers are busy. "$\mathbf{N_i+1}$" is a macro-state corresponding to "congestion for priority i given congestion for priority $i+1$". Since we are about to define matrices whose indices vary from 1 to N_i+1 and correspond to states "1" through "$\mathbf{N_i+1}$", it is convenient to use the above notation. (We shall, whenever possible, use boldfaced characters for state "$\mathbf{N_i+1}$", to distinguish this state from states "m", where $1 \leq m \leq N_i$.)

With these definitions in mind, we propose to compute the expected number of visits to states "m" ($1 \leq m \leq N_i+1$), during a priority $i+1$ congested period (C_{i+1}). If we know the expected holding time in these states, we can easily compute the *conditional* steady state probability of an arrival finding the system in one of these states, conditional on the arrival occuring while the system is congested for priority $i+1$ (C_{i+1}). We then decondition on C_{i+1} to find $E[C_i]$, $E[U_i]$, p_i, q_i and $P_{n|U_i}$, which completes the inductive (recursive) derivation of the quantities of interest.

In order not to break the flow of the arguments of this section, we only present here the bare essentials of the derivation of the steady state results that we seek. For a detailed technical discussion and justification of these derivations the reader is referred to the appendix. With that remark , let us now turn to the steady state results.

How is a priority $i+1$ blocked period (C_{i+1}) initiated? Obviously, just before the transition that first congests the system for priority $i+1$, the system is in macro-state U_{i+1}. C_{i+1} is triggered by an arrival of priority 1 through $i+1$; arrivals of lower priority cannot, by definition of the congestion period, trigger C_{i+1}. (Note that if we had include the state "N_i servers busy and all queues of priority i or higher empty" in the congested macro-state, lower priority arrivals could have triggered $Ci+1$, which would have singularly complicated our derivations.) Now, a triggering arrival of priority higher than i has access to N_i (and possibly more) servers, while a triggering arrival of priority $i+1$ only has access to N_{i+1} servers. We must therefore distinguish two cases:

(i) C_{i+1} is triggered by an arrival of priority $i+1$, and

(ii) C_{i+1} is triggered by an arrival of priority 1 through i.

We define:

Definition 3.6: Triggering Probabilities

α_{in} is the **probability that C_{i+1} is triggered by an arrival of priority i or higher and that the first state reached in C_{i+1} is state "n"**, for $n \in \{N_{i+1}+1, \ldots, N_i+1\}$. (Recall that "n" is the state "n servers busy and system in C_{i+1}".)

β_{ilk} is the **probability that C_{i+1} is triggered by a priority $i+1$ arrival that arrives to state "l servers busy and system in U_{i+1}" and requests k servers.**

In the appendix, we show that

$$\alpha_{in} = \frac{\sum_{j=1}^{i} \lambda_j \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \sigma_{j,n-l}}{\sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \left(\sum_{j=1}^{i+1} \lambda_j \sum_{k=N_{i+1}-l+1}^{N_j} \sigma_{jk} \right)} \qquad for\, n \in \{N_{i+1}+1, \ldots, N_i\}, \tag{3.50}$$

$$\alpha_{i,N_i+1} = \frac{\sum_{j=1}^{i} \lambda_j \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \sum_{k=N_i-l+1}^{N_j} \sigma_{jk}}{\sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \left(\sum_{j=1}^{i+1} \lambda_j \sum_{k=N_{i+1}-l+1}^{N_j} \sigma_{jk} \right)}, \tag{3.51}$$

and,

$$\beta_{ilk} = \frac{\lambda_{i+1} P_{l|U_{i+1}} \sigma_{i+1,k}}{\sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \left(\sum_{j=1}^{i+1} \lambda_j \sum_{k=N_{i+1}-l+1}^{N_j} \sigma_{jk} \right)} \quad for\ l \in \{1, ..., N_{i+1}\}\ and\ k \in \{N_{i+1}-l+1, ..., N_{i+1}\}. \tag{3.52}$$

We now define:

Definition 3.7: Expected number of visits

Ξ_{in} **is the expected number of visits to state "n" during a congestion period** C_{i+1}.

η_{imn} **is the expected number of visits to state "n" during** an elementary delay cycle $R_{i,m}$, for $m \in \{1, 2, ..., N_i+1\}$ and $n \in \{1, 2, ..., N_i+1\}$. Define the (N_i+1)-by-(N_i+1) matrix $\underline{\mathbf{H}}_i$ as $(\underline{\mathbf{H}}_i)_{mn} \equiv \eta_{imn}$.

We show in the appendix that $\underline{\mathbf{H}}_i$ is determined by $\underline{\mathbf{H}}_i = \underline{\mathbf{I}} + \underline{\mathbf{A}}_i \cdot \underline{\mathbf{H}}_i$, where the matrix $\underline{\mathbf{A}}_i$ is determined by:

$$(\mathbf{A}_i)_{ml} \equiv \begin{cases} 0 & \text{for } m \in \{1, 2, ..., N_i+1\} \text{ and } l \in \{1, 2, ..., m-1\} \\ \sum_{k=l-m+1}^{N_i-m+1} \Delta_{imk} & \text{for } m \in \{1, 2, ..., N_i+1\} \text{ and } l \in \{m, ..., N_i+1\} \end{cases}, \tag{3.60}$$

with

$$\Delta_{imk} \equiv \frac{\sum_{r=1}^{i} \lambda_r \sigma_{rk}}{\lambda_i^c + m\mu} \quad \text{for } m \in \{1, 2, ..., N_i+1\} \text{ and } k \in \{1, 2, ..., N_i - m\}, \tag{3.56}$$

and,

$$\Delta_{i,m,N_i-m+1} \equiv \frac{\sum_{r=1}^{i} \lambda_r \sum_{k=N_i-m+1}^{N_r} \sigma_{rk}}{\lambda_i^c + m\mu} \quad \text{for } m \in \{1, 2, ..., N_i+1\}. \tag{3.57}$$

Using equations (3.50-52), Ξ_{in} is obtained from:

$$\Xi_{in} = \sum_{m=N_{i+1}+1}^{N_i+1} \alpha_{im} \sum_{l=N_{i+1}+1}^{m} \eta_{imn} + \sum_{l=1}^{N_{i+1}} \sum_{k=N_{i+1}-l+1}^{N_{i+1}} \beta_{ilk} \sum_{m=N_{i+1}-k+1}^{l} \eta_{imn}$$

$$+\lambda_{i+1}E[B_{i+1}]\sum_{k=1}^{N_{i+1}}\sigma_{i+1,k}\sum_{m=N_{i+1}-k+1}^{N_{i+1}}\eta_{imn}\ . \tag{3.64}$$

From the expected number of visits to states "n" (for $n\in\{1,2,...,N_i+1\}$) during a congestion period C_{i+1}, it is now easy to complete our recursions.

For $n\in\{1, 2, ..., N_i\}$, the expected holding time, $E[T_{in}]$, in state "n" , per visit, is given by $1/(\lambda_i^c+n\mu)$. For $n=N_i+1$, however, the expected time $E[T_{i,N_i+1}]$ spent in state "N_i+1" is not so easily computed. Notice especially that the time spent in state "N_i+1" depends on where the transition into state "N_i+1" is made from. While there may be ways of deriving the expected time spent in "N_i+1" directly, it is more expedient at this stage to make use of our knowledge of the expected duration of C_{i+1}. Indeed, the following identity holds:

$$E[C_{i+1}]=\sum_{n=1}^{N_i+1}\Xi_{in}E[T_{in}]=\sum_{n=1}^{N_i}\Xi_{in}\frac{1}{\lambda_i^c+n\mu}+\Xi_{i,N_i+1}E[T_{i,N_i+1}]\ , \tag{3.17}$$

from which one easily deduces $E[T_{i,N_i+1}]$:

$$E[T_{i,N_i+1}]=\frac{E[C_{i+1}]-\sum_{n=1}^{N_i}\Xi_{in}\frac{1}{\lambda_i^c+n\mu}}{\Xi_{i,N_i+1}}\ . \tag{3.18}$$

We are now able to compute the steady state probabilities $Q_{n|C_{i+1}}$ of being in state "n", given that the system is in macro-state C_{i+1}. They are given by

$$Q_{n|C_{i+1}}=\frac{\Xi_{in}E[T_{in}]}{\sum_{j=1}^{N_i+1}\Xi_{ij}E[T_{ij}]}\quad for\, n\in\{1, 2, ..., N_i+1\}. \tag{3.19}$$

Now define p_i as the probability of a random Poisson arrival finding the system in a non-busy period (for priority i), U_i , and q_i as the probability of finding it in a priority i busy period C_i. Notice that, by definition of state "N_i+1", $E[C_i]=E[T_{i,N_i+1}]$. A simple conditioning argument lets us write p_i and q_i as:

$$p_i=p_{i+1}+q_{i+1}\sum_{n=1}^{N_i}Q_{n|C_{i+1}}=p_{i+1}+(1-p_{i+1})(1-Q_{N_i+1|C_{i+1}})\quad and\quad q_i=1-p_i\ . \tag{3.20}$$

Finally, the probabilities $P_{n|U_i}$ of finding the system in state "n servers busy *given the system is unblocked for priority i*" are given by:

$$P_{n|U_i} = \frac{p_{i+1} P_{n|U_{i+1}} + q_{i+1} Q_{n|C_{i+1}}}{p_{i+1} \sum_{m=0}^{N_{i+1}} P_{m|U_{i+1}} + q_{i+1} \sum_{m=1}^{N_i} Q_{m|C_{i+1}}} \quad \text{for } n \in \{0, 1, 2, \dots, N_i\}. \tag{3.21}$$

To close the induction argument, we use equations (3.10), (3.18) and (3.20) to complete this section by the recursions for $E[U_i]$:

$$E[U_i] = \frac{p_i}{1-p_i} E[C_i] = \frac{p_i}{1-p_i} E[T_{N_i+1}] \tag{3.22}$$

With the steady state probabilities computed in this section, we now finally have the building blocks necessary for the derivation of the waiting time distributions of the various prioritized arrival streams.

3.4 WAITING TIME DISTRIBUTIONS

The waiting times for the various priorities can now be determined from the quantities derived in the preceding sections. We focus on a random (tagged) priority i customer.

3.4.1 FCFS within a Priority

- With probability p_i she arrives during U_i. Given that she arrives to an uncongested system, she has a probability $P_{n|U_i}$ of arriving while exactly n servers are busy, $n \in \{0, 1, ..., N_i\}$. Independently of anything else, she requires exactly k servers with probability σ_{ik}, and may therefore have to wait until a sufficient number of servers are idle. Her conditional waiting time distribution (in transform domain) will be equal to:

$$E\left[e^{-sW_i} \middle| \text{arrival requesting k servers during } U_i \text{ while n servers busy} \right] = \prod_{m=N_i-k+1}^{n} R^*_{i-1,m}(s) \tag{3.23}$$

- Alternatively, the tagged priority i customer may arrive during a blocked period, C_i. She must then wait in the priority i queue until she may enter service. This queue moves up at independent identically distributed intervals, *except* for the *first* (priority i) customer who arrives during a congestion period. The first queue move-up time is still independent of the others, but it is not governed by the same distribution:

A congestion period, C_i, can be started by arrivals of priorities 1 through i+1 requesting varying numbers of servers. In general, the probability that the triggering customer requests k servers, even if she is of priority i, is not equal to σ_{ik}, as evidenced by equation (3.26) below. On the other hand, priority i customers that arrive (and enter service) during C_i request k servers with probability σ_{ik}, independent of anything else. Therefore queue move-ups between these customers are independent, *identically* distributed, while the distribution from the beginning of a congested period until the *first* priority i customer could get served is in general distributed *differently*.

Definition 3.8: Queue Move-up Times

We define the **first queue move-up time**, F_i, as the duration of the **time interval that starts from the instant a** (priority i) **congestion period,** C_i, **begins, and ends with the first instant a** (random priority i) **customer could enter service.**

Similarly, we define a **regular queue move-up time, S_i**, as the duration of the **time interval that starts from the instant** some (priority i) **customer *that arrived during the congestion period*, C_i, enters service, until the first moment the next** (random priority i) **customer could enter service.**

By analogy with M/G/1 queues with exceptional first service time during a busy period, in the M/M/{N_i}⊗{$\underline{S}$} system, a random priority i customer that arrives to a congested system will experience a waiting time distributed as

$$E\left[e^{-sW_i} \,\middle|\, \text{arrival during a congestion period } (C_i) \right] = \frac{1-F_i^*(s)}{\left(s-\lambda_i+\lambda_i S_i^*(s) \right) E[C_i]} , \tag{3.24}$$

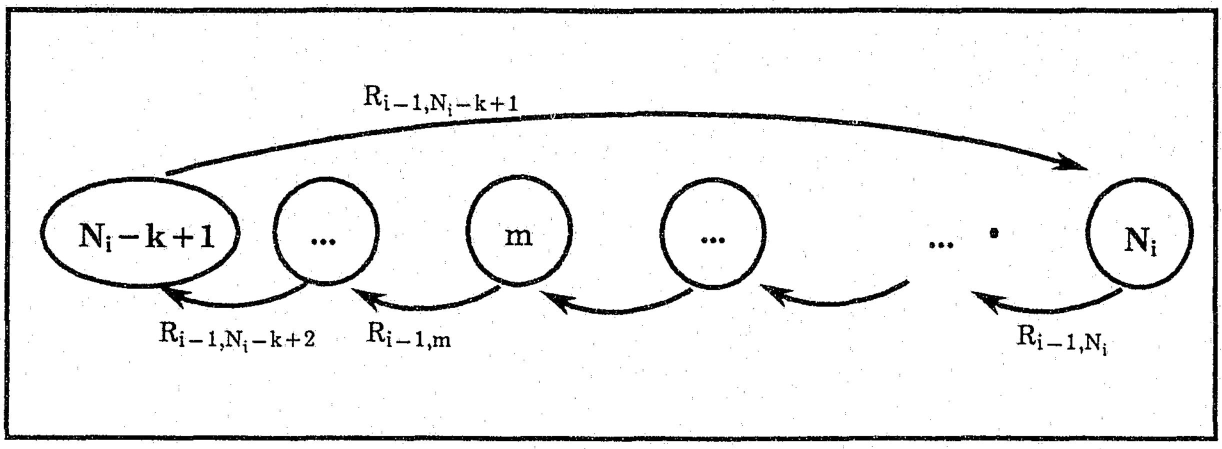
where $F_i^*(s)$ and $S_i^*(s)$ are, respectively, the transform of the first queue move-up time and the transform of a regular queue move-up time (for priority i) during the congestion period C_i.

The direct derivation of the above (Pollaczek-Khinchin) transform equation is shown in the appendix. We would like to emphasize the fact that, conditional upon arrival during a congestion period, our random arrival is confronted with an M/G/1 queue with (regular) service time S_i and a special first service time Fi at the beginning of the M/G/1 busy (congestion) period. Such systems have been studied extensively (e.g., Welch [1964]), notably in server vacation models

$S_i^*(s)$ is easily found (Figure 3.3) using delay cycles:

$$S_i^T(s) = \sum_{k=1}^{N_i} \sigma_{ik} \left(\prod_{m=N_i-k+1}^{N_i} R_{i-1,m}^*(s) \right) , \tag{3.25}$$

$F_i^*(s)$ is a little more complicated since this quantity depends on who initiates the blocked period. The probability that a congestion period is initiated by a priority j customer (j≤i) requesting k servers arriving to state "n servers busy, *given* system uncongested", is given by:



– Figure 3.3 –

$$\frac{P_{n|U_i}\lambda_j\sigma_{jk}}{\sum_{m=0}^{N_i} P_{m|U_i}\sum_{r=1}^{i}\lambda_r\sum_{l=N_i-m+1}^{N_j}\sigma_{jl}}\quad,\quad \text{for } n\in\{0,2,\ldots,N_i\},\ k\in\{N_i-n+1,\ldots,N_j\} \text{ and } j\in\{1,\ldots,i\}$$

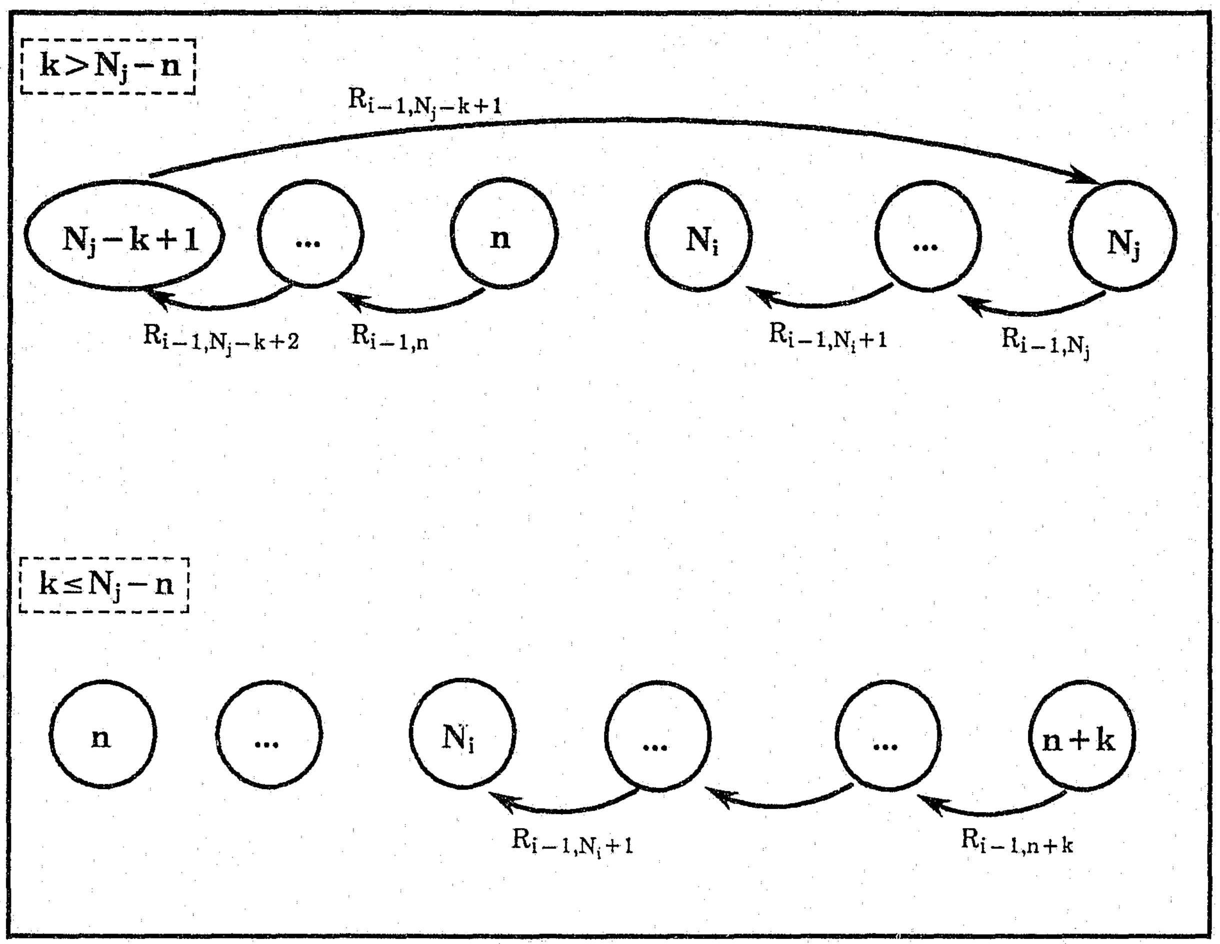
Unconditioning, we can write (see **Figure 3.4**):

$$F_i^T(s)=\frac{\sum_{n=0}^{N_i} P_{n|U_i}\sum_{j=1}^{i}\lambda_j\sum_{k=N_i-n+1}^{N_j}\sigma_{jk}\left(\prod_{m=N_j-k+1}^{n} R^T_{i-1,m}(s)\right)\left(\prod_{m=N_i+1}^{\min(N_j,n+k)} R^T_{i-1,m}(s)\right)}{\sum_{m=0}^{N_i} P_{m|U_i}\sum_{r=1}^{i}\lambda_r\sum_{l=N_i-m+1}^{N_j}\sigma_{jl}} \tag{3.26}$$

Finally, we obtain for the transform of the distribution of the waiting time, W_i:

$$W_i^*(s)=E\left[e^{-sW_i}\right]=p_i\sum_{n=0}^{N_i} P_{n|U_i}\sum_{k=1}^{N_i}\sigma_{ik}\left(\prod_{m=N_i-k+1}^{n} R^*_{i-1,m}(s)\right)+q_i\frac{1-F_i^*(s)}{\left(s-\lambda_i+\lambda_i S_i^*(s)\right)E[C_i]}\ . \tag{3.27}$$

Note that, for $k\leqq C_i-n$, the product (Π) in the above expression reduces to 1, by our default convention. We find that the waiting time distribution does indeed exhibit the expected impulse at zero:



– Figure 3.4 –

$$\mathrm{Prob}(W_i = 0) = p_i \sum_{n=0}^{N_i - 1} P_{n|U_i} \sum_{k=1}^{N_i - n} \sigma_{ik} \qquad (3.28)$$

3.5 Stability

We have hitherto implicitly assumed that the system analyzed is stable, in the sense that, for all priorities, the expected waiting times are finite. We now address the stability issue in more detail. Apart from global system stability, there are, in general, for the M/M/{N_i}⊗{$\underline{\mathrm{S}}$} system, stability conditions for each individual priority stream: finite expected waiting times for each stream. Because of the HOL service discipline, it is clear that if the system is unstable for priority i, it is necessarily unstable for priority j, where j>i.

Assume the system is stable for priority $i-1$. A **necessary condition for stability** up to priority i is given by

$$\lambda_i E\left[S_i\right] = \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right] < 1 \quad . \tag{3.29}$$

This condition is clearly necessary, since it merely requires that during a random (regular) priority i queue move-up time there occur less than one arrival on average. This condition is the typical (M/G/1) condition that the utilization factor of the (here virtual) server be less than 1.

We now show, more rigorously, that this condition is both necessary and sufficient.

If the system is unstable, equations (3.6), (3.7) and (3.9) still hold. The way equation (3.8) was derived, however, it implicitly assumes that

$$\lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0) = 0 \, ,$$

and it must therefore be given special attention when the system is unstable: The **stability** of the M/M/{N_i}⊗{$\underline{\mathrm{S}}$} system depends on the mathematical stability of the delay cycle equations:

$$R^*_{i,N_i}(s) = E\left[e^{-sR_{i,N_i}}\right] = R^*_{i-1,N_i}\left(s + \lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(s)\right) \tag{3.6}$$

Stability up to priority $i-1$ requires that

$$E\left[R_{i-1,n}\right] < \infty \ , \ \text{for } 1 \le n \le N \ .$$

By differentiation of equation (3.6), we obtain, in general:

$$E\left[R_{i,n}\right] = -\frac{dR^*_{i-1,n}}{dx}(\lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0)) \cdot \left(1+\lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right]\right) \quad (3.30)$$

Forming a weighted sum of these equations:

$$\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] = -\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} \left[\frac{dR^*_{i-1,n}}{dx}(\lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0))\right.$$

$$\left. \cdot \left(1+\lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right]\right)\right], \quad (3.31)$$

whence:

$$\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] = \frac{-\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} \frac{dR^*_{i-1,n}}{dx}(\lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0))}{1+\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} \frac{dR^*_{i-1,n}}{dx}(\lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0))}. \quad (3.32)$$

Because the system is stable for $i-1$, we can write,by continuity at $\lambda_i = 0$:

$$\exists \Lambda > 0, \forall \lambda_i < \Lambda,\ \lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0) = 0 \ . \quad (3.33)$$

In other words, $\exists \Lambda > 0$, $\forall \lambda_i < \Lambda$, the system is stable for priority i. Equation(3.32), for $\lambda_i < \Lambda$, can be rewritten as:

$$\exists \Lambda > 0, \forall \lambda_i < \Lambda,\ \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] = \frac{-\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} \frac{dR^*_{i-1,n}}{dx}(0)}{1+\lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} \frac{dR^*_{i-1,n}}{dx}(0)},$$

or:

$$\exists \Lambda > 0, \forall \lambda_i < \Lambda,\ \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] = \frac{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]}{1-\lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]}. \quad (3.34)$$

Therefore,

$$\exists \Lambda > 0, \forall \lambda_i < \Lambda, \ \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] < \infty \tag{3.35}$$

and:

$$\Lambda \le \frac{1}{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]} . \tag{3.36}$$

which proves our necessary condition.

Now, let us prove sufficiency.

Let Λ_{max} be defined as:

$$\Lambda_{max} = \max\left\{\Lambda > 0 \ \middle|\ \forall \lambda_i < \Lambda, \ \lambda_i - \lambda_i \sum_{j=1}^{N_i} \sigma_{ij} \prod_{n=N_i-j+1}^{N_i} R^*_{i,n}(0) = 0\right\} . \tag{3.37}$$

Suppose

$$\Lambda_{max} < \frac{1}{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]} . \tag{3.38}$$

By definition of Λ_{max},the system is unstable for $\lambda_i = \Lambda_{max}$:

$$\text{for } \lambda_i = \Lambda_{max} , \ \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] = \infty . \tag{3.39}$$

But, from equation (3.34), we know that,

$$\text{for } \lambda_i < \Lambda_{max} , \ \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i,n}\right] < \frac{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]}{1 - \Lambda_{max} \sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]} < \infty . \tag{3.40}$$

The upper bound in equation (3.40) is finite and independent of λ_i, yet $E[R_{i,n}]$ is a continuous function of λ_i; thus we arrive at a contradiction. Therefore, our original hypothesis (3.38) is false, and:

$$\Lambda_{max} \ge \frac{1}{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]} . \tag{3.41}$$

Finally, combining equations (3.36) and (3.41), we conclude that:

$$\Lambda_{max} = \frac{1}{\sum_{j=1}^{N_i} \sigma_{ij} \sum_{n=N_i-j+1}^{N_i} E\left[R_{i-1,n}\right]}, \tag{3.42}$$

or, equivalently, that equation (3.29) is a necessary and sufficient stability condition.

4 Loss Systems

It is easy to extend the results for the M/M/$\{N_i\}\otimes\{\underline{S}\}$ model to systems where customers of certain priorities are lost if they arrive when the system is congested.

Assume that priority i customers are lost if the system is congested for priority i. Then equation (3.6) must be modified to:

$$R^*_{i,n}(s) = R^*_{i-1,n}(s) \qquad \text{for } N_i \le n \le N. \tag{4.1}$$

Similarly, equation (3.8) becomes:

$$E\left[R_{i,n}\right] = E\left[R_{i-1,n}\right] \qquad \text{for } N_i \le n \le N, \tag{4.2}$$

These equations are used for the recursions defining the $R_{i,n}$'s.

The steady state probabilities are computed from the expected number of visits, η_{imn}, to state "n" during a delay cycle $R_{i,m}$. These η_{imn}'s have to be appropriately modified if customers arriving to a congested system are lost:

$$\eta_{imn} = \delta_{mn} + \sum_{k=1}^{N_i-m+1} \Delta'_{imk} \sum_{l=m}^{m+k} \eta_{ln} \qquad \begin{cases} \text{for } m\in\{1, 2, \dots, N_i+1\} \\ \text{and } n\in\{1, 2, \dots, N_i+1\} \end{cases} \tag{4.3}$$

where δ_{ij} is Kronecker's delta, and Δ'_{imk} is the probability that the first transition from state "m servers busy" is caused by an arrival (of priority $i-1$ or higher) requesting k servers. Δ'_{imk} is defined by:

$$\Delta'_{imk} \equiv \frac{\sum_{r=1}^{i-1} \lambda_r \sigma_{rk}}{\lambda^c_{i-1} + m\mu} \qquad \text{for } k \in \{1, 2, \ldots, N_i - m + 1\}\ , \tag{4.4}$$

and,

$$\Delta'_{i,m,N_i-m+1} \equiv \frac{\sum_{r=1}^{i-1} \lambda_r \sum_{k=N_i-m+1}^{N_r} \sigma_{rk}}{\lambda^c_{i-1} + m\mu}\ . \tag{4.5}$$

Note that the loss system described here only loses customers that arrive during a congestion period. We have assumed that a customer that arrives to an uncongested system gets served, even though she may have to wait until sufficient servers become available. One can, of course make the assumption that a priority i customers that arrives to an uncongested system is also lost, if she requests more servers than are available upon arrival. The arguments about initiations of congestion periods can be easily adjusted for this alternative (which we shall not treat, here).

5 Concluding remarks on the $M/M/\{N_i\}\otimes\{\underline{S}\}$ System

All important steady state probabilities derived in Section 3 (and Section 4) can be computed by solving (invertible) linear systems: the mathematical complexity is thus minimal. The heaviest calculations are matrix inversions of matrices of size on the order of N-by-N; for many practical applications, $N \le 25$, so the computational burden is not very heavy. The only painful part is the setting up of the bounds on the (sometimes triple) summations that abound throughout the derivations.

In this paper, we have presented a methodology for solving a moderately complex queueing model by an M/G/1 based decomposition approach, where classical solution procedures based on global balance equations and discrete transform techniques would have dismally failed. The basic $M/M/\{N_i\}\otimes\{\underline{S}\}$ system presented in this paper is but one of a family of models that can be tackled in a similar fashion:

A conceptually simple extension of the $M/M/\{N_i\}\otimes\{\underline{S}\}$ system is the following (proposed by *Green* [1984]): Assume every arriving customer arrives, not with a server requirement of k servers, but with a **requirement of the form (s,S)** meaning that the customer wants S servers if the system is not too busy, but she will do with s, $s+1$, ..., or $S-1$ if necessary. This modification changes the coefficients of certain equations and matrices of Section 3, but the general argument remains valid.

The case with **non-preemption during the assignment phase,** *Green* [1984]'s version of the stochastic-server-requirements problem, briefly alluded to in Section 2.2, above) is similarly tractable by the M/G/1 decomposition method.

Other extensions and hybrid policies pose no major theoretical or computational problems. The reader is referred to *Schaack* [1985] for a detailed discussion of some such generalizations. We believe the family of cutoff models developed there, of which the basic $M/M/\{N_i\}\otimes\{\underline{S}\}$ model is a prime representative, will offer a tool for evaluating a whole range of interesting and useful policy alternatives for prioritized service environments.

APPENDIX

Appendix A.1 contains the step-by-step derivations of some intermediate results that enable us to compute the steady state results of Section 3.3.2. These derivations were relegated to the appendix, so as not to overburden the main body of the text with an excessive number of definitions and equations. Appendix A.2 derives the Pollaczek-Khinchin waiting time transform formulas for the $M/M/\{N_i\}\otimes\{\underline{S}\}$ queue.Appendix A.3 contains, for quick reference, a table of the major definitions used throughout Section 3.

A.1 The Expected Number of Visits to States "m"

We focus here on priority i congested and uncongested periods, and we derive some intermediate results for the recursive arguments of Section 3.3.2).

We obtain the expected number of visits to states "m", with $1 \leq m \leq N_i + 1$. We recall (from Definition 3.5 in Section 3.3.2) that "m" is defined as "m servers are busy and the system is uncongested for priority i, *given* that the system is congested for priority i + 1" (i.e., $U_i \wedge m | C_{i+1}$), for $1 \leq m \leq N_i$, and "$N_i + 1$" is defined as "the system is congested for priority i, *given* that it is congested for priority i + 1" (i.e., $C_i | C_{i+1}$).

In order to simplify the derivations, we further use the definitions of the triggering probabilities introduced in Section 3.3.2 (Definition 3.6):

α_{in}	$\equiv$	probability that C_{i+1} is triggered by an arrival of priority i or higher and the first state reached in C_{i+1} is state "n"; for $n \in \{N_{i+1}+1, ..., N_i+1\}$.
β_{ilk}	$\equiv$	probability that B_{i+1} is triggered by a priority i + 1 arrival that arrives to state "*l servers busy and system in* U_{i+1}" and requests k servers.

The probabilities α_{in} and β_{ilk} are obtained directly from the conditional probabilities $P_{n|U_{i+1}}$ and the server requirements $\underline{S}$, by appropriate conditioning:

$$\alpha_{in} \propto \sum_{j=1}^{i} \lambda_j \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \sigma_{j,n-l} \qquad , \quad for\, n \in \{C_{i+1}+1, ..., C_i\},$$

$$\alpha_{i,N_i+1} \propto \sum_{j=1}^{i} \lambda_j \sum_{l=0}^{C_{i+1}} P_{l|U_{i+1}} \sum_{k=C_i-l+1}^{C_j} \sigma_{jk} \quad .$$

And,

$$\beta_{ilk} \propto \lambda_{i+1} P_{l|U_{i+1}} \sigma_{i+1,k} \qquad \text{for } l \in \{1, 2, \ldots, N_{i+1}\} \text{ and } k \in \{N_{i+1} - l + 1, \ldots, N_{i+1}\}.$$

The constant of proportionality is found to be the inverse of

$$\Pi \equiv \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \left(\sum_{j=1}^{i+1} \lambda_j \sum_{k=N_{i+1}-l+1}^{N_j} \sigma_{jk} \right) \tag{3.49}$$

So:

$$\alpha_{in} = \frac{1}{\Pi} \sum_{j=1}^{i} \lambda_j \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \sigma_{j,n-l} \qquad \text{for } n \in \{N_{i+1}+1, \ldots, N_i\}, \tag{3.50}$$

$$\alpha_{i,N_i+1} = \frac{1}{\Pi} \sum_{j=1}^{i} \lambda_j \sum_{l=0}^{N_{i+1}} P_{l|U_{i+1}} \sum_{k=N_i-l+1}^{N_j} \sigma_{jk} \quad , \tag{3.51}$$

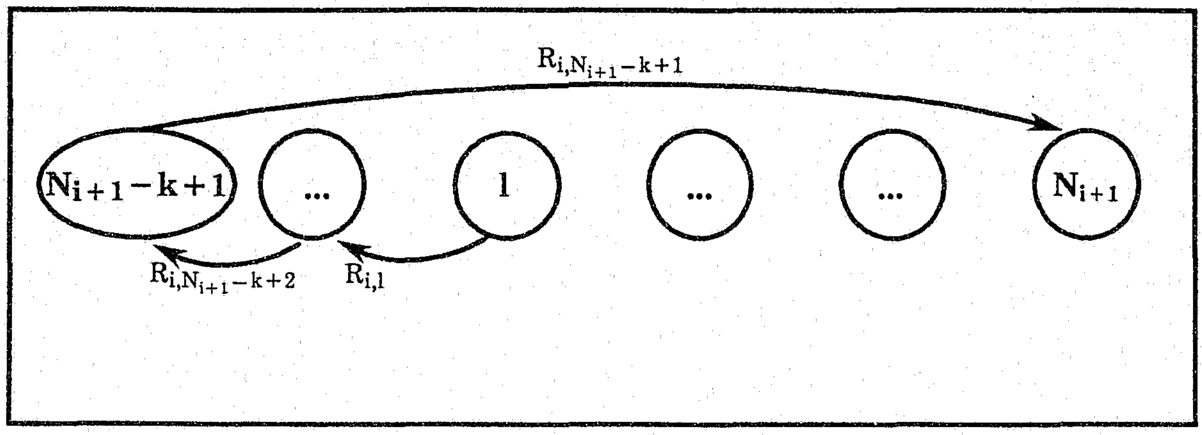
$$\beta_{ilk} = \frac{1}{\Pi} \lambda_{i+1} P_{l|U_{i+1}} \sigma_{i+1,k} \qquad \text{for } l \in \{1, 2, \ldots, N_{i+1}\} \text{ and } k \in \{N_{i+1} - l + 1, \ldots, N_{i+1}\}. \tag{3.52}$$

Now that we know how C_{i+1} is triggered, we ask the question: What is the expected number of visits, ξ_{in}, to state "n" from the moment the system enters C_{i+1}, until absorption in state "N_{i+1} servers busy, *priority i unblocked and triggering customer has started service*"? (This moment of absorption always occurs on a downward transition, for the system is skipfree negative on the state space {"1", "2", ..., "$N_i + 1$"}.) This moment of absorption can be viewed as the time instant at which the system would drop into macro-state U_{i+1} again **had there occurred no priority i+1 arrival during C_{i+1}**. (For simplicity, assume therefore that the priority i+1 arrival stream is temporarily suppressed. We shall restore it shortly.)

- **Suppose C_{i+1} is triggered** from state "l servers busy and system in macro-state U_{i+1}" **by a priority i+1 arrival** requesting k servers. Then the time until absorption in state "C_{i+1} servers busy, *priority i unblocked and triggering customer has started service*" is given simply by the time it takes to start service on the triggering (priority i + 1) arrival. This time is distributes as

$$\sum_{m=N_{i+1}-k+1}^{l} R_{i,m} \ , \tag{3.53}$$

as **Figure 3.5** illustrates.Therefore ξ_{in} is given by



– Figure 3.5 –

C_{i+1} triggered by a priority $i+1$ customer requesting k servers ($i<T$).

$$\sum_{m=N_{i+1}-k+1}^{l} \eta_{imn} , \tag{3.54}$$

where η_{imn} is the expected number of visits to "n" (excluding the last transition to $m-1$ servers busy) during $R_{i,m}$.

The transient process is skip-free negative on the state space {"1", "2", ..., "N_i+1"}; therefore we can write, conditioning on the first transition from state "m servers busy":

$$\eta_{imn} = \delta_{mn} + \sum_{k=1}^{N_i-m+1} \Delta_{imk} \sum_{l=m}^{m+k} \eta_{ln} \qquad \begin{cases} \text{for } m \in \{1, 2, ..., N_i+1\} \\ \text{and } n \in \{1, 2, ..., N_i+1\} \end{cases} \tag{3.55}$$

where δ_{ij} is Kronecker's delta, and Δ_{imk} is the probability that the first transition from state "m servers busy" is caused by an arrival (of priority i or higher) requesting k servers, for $k<N_i-m+1$; and, for $k=N_i-m+1$, Δ_{imk} is the probability that the first transition from state "m servers busy" is caused by an arrival (of priority i or higher) requesting at least N_i-m+1 servers.

Note that Δ_{i,m,N_i-m+1} is the transition probability from state "m" into superstate "N_i+1", i.e. into a priority i congestion period, C_i.

Δ_{imk} is given by:

$$\Delta_{imk} \equiv \frac{\sum_{r=1}^{i} \lambda_r \sigma_{rk}}{\lambda_i^c + m\mu} \qquad \text{for } k \in \{1, 2, ..., N_i-m+1\} , \tag{3.56}$$

and,

$$\Delta_{i,m,N_i-m+1} \equiv \frac{\sum_{r=1}^{i} \lambda_r \sum_{k=N_i-m+1}^{N_r} \sigma_{rk}}{\lambda_i^c + m\mu} \tag{3.57}$$

Equations (3.55) can be rewritten as

$$\eta_{imn} = \delta_{mn} + \sum_{l=m}^{N_i+1} \left(\sum_{k=l-m+1}^{N_i-m+1} \Delta_{imk} \right) \eta_{iln} \qquad \begin{cases} \text{for } m \in \{1, 2, ..., N_i+1\} \\ \text{and } n \in \{1, 2, ..., N_i+1\} \end{cases} \tag{3.58}$$

One recognizes a linear system of the form $\underline{\mathbf{H}}_i = \underline{\mathbf{I}} + \underline{\mathbf{A}}_i \cdot \underline{\mathbf{H}}_i$, where $\underline{\mathbf{H}}_i$ and $\underline{\mathbf{A}}_i$ are matrices defined by:

$$(\mathbf{H}_i)_{mn} \equiv \eta_{imn} \qquad \text{for } (m,n) \in \{1, 2, ..., N_i+1\}^2 \tag{3.59}$$

and:

$$(\mathbf{A}_i)_{ml} \equiv \begin{cases} 0 & \text{for } m \in \{1, 2, ..., N_i+1\} \text{ and } l \in \{1, 2, ..., m-1\} \\ \sum_{k=l-m+1}^{C_i-m+1} \Delta_{imk} & \text{for } m \in \{1, 2, ..., N_i+1\} \text{ and } l \in \{m, ..., N_i+1\} \end{cases} \tag{3.60}$$

$\mathbf{I} - \mathbf{A}$ is an upper triangular matrix of full rank and therefore invertible.

All diagonal elements of $\underline{\mathbf{A}}_i$ are between 0 and 1, therefore all eigenvalues of $\underline{\mathbf{A}}_i$ are between 0 and 1; thus $\underline{\mathbf{A}}_i^n \to \infty$ as $n \to \infty$, and: $\underline{\mathbf{H}}_i = (\underline{\mathbf{I}} - \underline{\mathbf{A}}_i)^{-1} = \underline{\mathbf{I}} + \underline{\mathbf{A}}_i + \underline{\mathbf{A}}_i^2 + \underline{\mathbf{A}}_i^3 + ...$ exists. (Since $\underline{\mathbf{A}}_i$ is a non-negative matrix, $\underline{\mathbf{H}}_i$ is non-negative: $\underline{\mathbf{H}}_i$ indeed yields sensible values for the expected number of visits η_{imn}.)

So far, we have only considered the case where the priority $i+1$ congestion period C_{i+1} is triggered by an arrival of priority $i+1$.

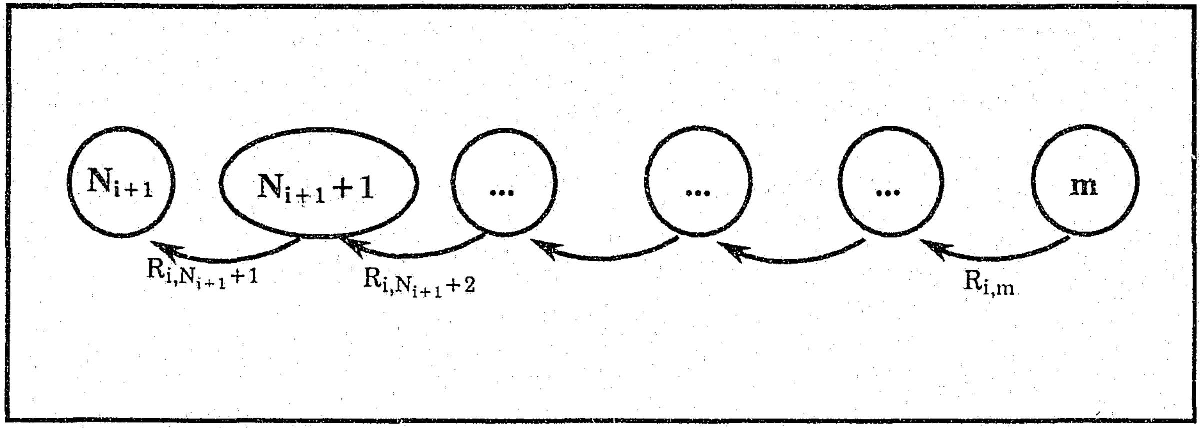
- Now **suppose C_{i+1} is triggered by an arrival of priority 1 through i.** Assume the first state reached in C_{i+1} is state "m". The time until absorption in state "C_{i+1} servers busy, *priority i unblocked and triggering customer has started service*" is given simply by the time it takes the system (recall that priority $i+1$ is temporarily suppressed) to drop down to state "N_{i+1}". The conditional value of ξ_{in} is thus given by

$$\sum_{l=N_{i+1}+1}^{m} \eta_{imn} \, . \tag{3.61}$$

as **Figure 3.6** illustrates.

- Unconditioning on the triggering event, we can now write, using equations (3.50-52), (3.54) and (3.61), for $n \in \{1, 2, ..., N_i+1\}$:

$$\xi_{in} = \sum_{m=N_{i+1}+1}^{N_i+1} \alpha_{im} \sum_{l=N_{i+1}+1}^{m} \eta_{imn} + \sum_{l=1}^{N_{i+1}} \sum_{k=N_{i+1}-l+1}^{N_{i+1}} \beta_{ilk} \sum_{m=N_{i+1}-k+1}^{l} \eta_{imn} \quad . \tag{3.62}$$



– Figure 3.6 –

C_{i+1} triggered by a priority j customer requesting k servers ($j \leq i < T$).

• Up to this point we have assumed that the priority $i+1$ arrival stream was turned off once C_{i+1} had been triggered. By definition, ξ_{in} counts the number of visits to state "n" in C_{i+1}, from the moment C_{i+1} is triggered until the system traps in state "C_{i+1} servers busy, *priority i unblocked and triggering customer has started service*". If there occurred no priority $i+1$ arrivals since C_{i+1} was triggered, the system would upon the last transition counted in $\xi_{i,N_{i+1}}$ have left macro-state C_{i+1} for macro-state U_{i+1}; that is, the system would become uncongested for priority $i+1$ upon this last transition. On the other hand, if there did occur one (or more) priority $i+1$ arrivals during C_{i+1}, upon the last transition to state "N_{i+1} servers busy" counted in $\xi_{i,N_{i+1}}$, the congestion period C_{i+1} would continue. We therefore now **restore the suppressed priority $i+1$ arrival stream.** During C_{i+1}, there arrive an expected $\lambda_{i+1}E[B_{i+1}]$ priority $i+1$ customers. Depending on the number of servers requested, each of these customers will contribute an expected number of visits to state "n" equal to

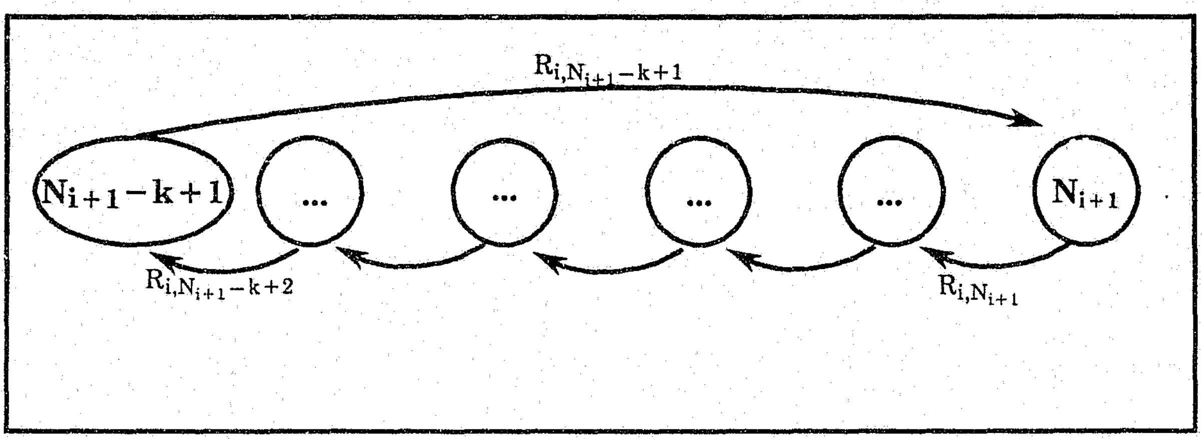
$$\sum_{m=N_{i+1}-k+1}^{N_{i+1}} \eta_{imn} \quad , \text{ where k is the number of servers requested by the customer,}$$

as illustrated in **Figure 3.7.**

Therefore, a random priority $i+1$ customer contributes

$$\sum_{k=1}^{N_{i+1}} \sigma_{i+1,k} \sum_{m=N_{i+1}-k+1}^{N_{i+1}} \eta_{imn} \tag{3.63}$$

visits to state "n". Define Ξ_{in} as the expected number of visits to state "n" in a



– Figure 3.7 –

C_{i+1} sustained by a priority i+1 customer requesting k servers (i<T).

priority i + 1 congestion period, C_{i+1}. Then one may write, for $n \in \{1, 2, ..., N_i + 1\}$:

$$\Xi_{in} = \xi_{in} + \lambda_{i+1} E[B_{i+1}] \sum_{k=1}^{N_{i+1}} \sigma_{i+1,k} \sum_{m=N_{i+1}-k+1}^{N_{i+1}} \eta_{imn} \quad ,$$

or, using equation (3.62),

$$\Xi_{in} = \sum_{m=N_{i+1}+1}^{N_i+1} \alpha_{im} \sum_{l=N_{i+1}+1}^{m} \eta_{imn} + \sum_{l=1}^{N_{i+1}} \sum_{k=N_{i+1}-l+1}^{N_{i+1}} \beta_{ilk} \sum_{m=N_{i+1}-k+1}^{l} \eta_{imn}$$

$$+ \lambda_{i+1} E[B_{i+1}] \sum_{k=1}^{N_{i+1}} \sigma_{i+1,k} \sum_{m=N_{i+1}-k+1}^{N_{i+1}} \eta_{imn} \quad . \tag{3.64}$$

This concludes the derivation of the expected number of visits, Ξ_{in}, to states "n" (for $n \in \{1, 2, ..., N_i + 1\}$) during a congestion period, C_{i+1}. In Section 3.3.2, these Ξ_{in}'s are used to derive the steady state probabilities p_i, q_i and $P_{n|U_i}$.

A.2 Conditonal Pollaczek-Khinchin Waiting Time Transform Formula (FCFS)

The duration of the time period, F_i, from the initiation of a congestion period (for priority i) to the time instant when the first priority i customer (arriving after the beginning of the congestion period) could be served has a distribution that is different from the queue move-up times experienced by subsequent arrivals to the priority i queue. This situation is analoguous to what happens in an M/G/1 queue in which the first customer served during a busy period experiences a service time distribution different from the one experienced by all other customers served during the busy period. Results for the M/G/1 case can be found in various places in the literature. The argumentation presented here closely parallels Kleinrock [1975, pp.219ff.].

The waiting time distribution for an arrival to a congested system is obtained in the following way. Consider a congestion period, C_i, for priority i. Let X_0 denote the first queue move-up time of the congestion period. All those customers who arrive during X_0 are served during the next interval whose duration is X_1. X_1 is the sum of the queue move-up times of all priority i customers who arrive during X_0. Similarly, at the expiration of X_1, all priority i customers who have arrived during X_1 get served during the next interval X_2. And so on, from X_t to X_{t+1}. We know that, if the system is stable, with probability one, there is a $\tau>0$, such that there are no priority i arrivals during X_τ. Since C_i denotes the total duration of the busy period, we have:

$$C_i = \sum_{t=0}^{\infty} X_t \ .$$

Conditioning on the duration of X_{t-1} and on N_{t-1}, the number of priority i arrivals during X_{t-1}, we can write:

$$E\left[e^{-sX_t} \;\middle|\; X_{t-1}=y, N_{t-1}=n\right] = [S_i^*(s)]^n \ .$$

Unconditioning successively on N_{t-1}, and then on X_{t-1}, we obtain:

$$E\left[e^{-sX_t} \;\middle|\; X_{t-1}=y\right] = e^{-(\lambda_i-\lambda_i S_i^*(s))\,y} \ ,$$

$$X_t^*(s) \equiv E\left[e^{-sX_t}\right] = X_{t-1}^*(\lambda_i-\lambda_i S_i^*(s)) \qquad \textbf{(3.65)}$$

Now, let's look at a (tagged) priority i customer who arrives during the congestion period. Suppose she arrives during X_t. Moreover assume X_t has a residual life Y_t and M_t priority i arrivals have already occured during X_t prior to our tagged arrival. Then we may write, for the waiting time of our new arrival:

$$E\left[e^{-sW_i}\ \middle|\ X_t=y, Y_t=y', N_t=n\right] = e^{-sy'}[S_i^*(s)]^n \quad .$$

Successive unconditioning yields:

$$E\left[e^{-sW_i}\ \middle|\ X_t=y, Y_t=y'\right] = e^{-sy'-(\lambda_i-\lambda_i S_i^*(s))(y-y')}$$

$$E\left[e^{-sW_i}\ \middle|\ X_t=y\right] = e^{-(\lambda_i-\lambda_i S_i^*(s))y}\ \frac{e^{-(s-\lambda_i+\lambda_i S_i^*(s))y}-1}{-(s-\lambda_i+\lambda_i S_i^*(s))\ y} = -\frac{e^{-sy}-e^{-(s-\lambda_i+\lambda_i S_i^*(s))y}}{(s-\lambda_i+\lambda_i S_i^*(s))\ y}$$

$$E\left[e^{-sW_i}\ \middle|\ \text{incidence into } X_t\right] = -\frac{X_t^*(s)-X_t^*(\lambda_i-\lambda_i S_i^*(s))}{[s-\lambda_i+\lambda_i S_i^*(s)]\,E\left[X_t\right]} \quad ,$$

or, using (3.65),

$$E\left[e^{-sW_i}\ \middle|\ \text{incidence into } X_t\right] = \frac{X_{t+1}^*(s)-X_t^*(s)}{[s-\lambda_i+\lambda_i S_i^*(s)]\,E\left[X_t\right]} \quad . \tag{3.66}$$

Now,

$$\text{Prob}\left[\text{incidence into } X_t\ \middle|\ \text{incidence into congestion period}\right] = \frac{E\left[X_t\right]}{E\left[C_i\right]} \quad . \tag{3.67}$$

Thus, unconditioning on t, and noting that X_0 is equal to F_i, we find the conditional Pollaczek-Khinchin transform equation:

$$E\left[e^{-sW_i}\,|\,\text{incidence into congestion period}\right] = \frac{1-X_0^*(s)}{[s-\lambda_i+\lambda_i S_i^*(s)]\,E\left[C_i\right]} = \frac{1-F_i^*(s)}{[s-\lambda_i+\lambda_i S_i^*(s)]\,E\left[C_i\right]} \quad . \tag{3.68}$$

A.3 Tables of Definitions

• T	number of priorities.
• N	total number of servers.
• N_i	server cutoff for priority i.
• λ_i	Poisson arrival rate for priority i.
• λ_i^c	cumulative arrival rate for priorities 1 through i.
• μ	exponential service rate.
• σ_{ik}	probability that a priority i customer requests k servers.
• $\underline{S}$	T-by-N matrix: $(\underline{S})_{ik} \equiv \sigma_{ik}$.
• $R_{0,n}$	time until first service completion after time t, when n servers are busy at t.
• $R_{i,n}$	elementary delay cycle from state "n servers busy and all queues (of priority 1 through i) empty", in a system with arrival streams of priorities $i+1$ through T suppressed (cf. also Table A.2).
• $FPT_{i,n,m}$	first passage time from state "n servers busy and all queues (of priority 1 through i) empty" to state "m servers busy and all queues (of priority 1 through i) empty", in a system with arrival streams of priorities $i+1$ through T suppressed.
• $V_{i,n}$	time until next transition form state"n servers busy", in a system with arrival streams of priorities $i+1$ through T suppressed.
• $X^*(s)$	Laplace-Stieltjes transform of the distribution of the random variable X.
• C_i	congestion state (or period) for priority i: at least one queue of priority i or higher is nonempty, or more than N_i servers are busy.
• U_i	uncongested state (or period) for priority i: all queues of priority 1 through i are empty and at most N_i servers are busy".
• p_i	steady state probability that the system is in state C_i.
• q_i	steady state probability that the system is in state U_i.
• $P_{n\|U_i}$	probability that there are n servers busy, given that the system is uncongested for priority i, where $n \in \{0, 1, ..., N_i\}$.

Table A.1 – Definitions

• "m"	state "m servers busy and the system is in macro-state U_i, given that the system is in macro-state C_{i+1}" for $1 \leq m \leqq N_i$.
• "N_i+1"	state "the system is in macro-state C_i, given that it is in macro-state C_{i+1}".
• α_{in}	probability that C_{i+1} is triggered by an arrival of priority i or higher and that the first state reached in C_{i+1} is state "n", for $n \in \{N_{i+1}+1, ..., N_i+1\}$.
• β_{ilk}	probability that C_{i+1} is triggered by a priority i+1 arrival that arrives to state "l servers busy and system in U_{i+1}" and requests k servers.
• Δ_{imk}	probability that the first transition from state "m servers busy" is caused by an arrival (of priority i or higher) requesting k servers, for $k < N_i - m + 1$; and, for $k = N_i - m + 1$, Δ_{imk} is the probability that the first transition from state "m servers busy" is caused by an arrival (of priority i or higher) requesting at least $N_i - m + 1$ servers.
• ξ_{in}	number of visits to state "n" from the moment the system enters C_{i+1}, until absorption in state "N_{i+1} servers busy, *priority i unblocked and triggering customer has started service*".
• Ξ_{in}	expected number of visits to state "n" during a congestion period C_{i+1}.
• η_{imn}	expected number of visits to state "n" during an elementary delay cycle $R_{i,m}$, for $m \in \{1, 2, ..., N_i+1\}$ and $n \in \{1, 2, ..., N_i+1\}$.
• $\underline{H}_i$	(N_i+1)-by-(N_i+1) matrix: $(\underline{H}_i)_{mn} \equiv \eta_{imn}$.
• T_{im}	expected holding time in state "m" , per visit.
• W_i	waiting time of a random priority i customer.
• S_i	regular queue move-up time.
• F_i	exceptional first queue move-up time in a congestion period.

Table A.1 (cont.) – Definitions

- **$R_{i,n}$ Elementary delay cycle from (n; 0, 0, 0, ...,0) in a system with arrival streams of priorities $i+1$ through T suppressed.**

  - All arrival streams of priority $i+1$ through T suppressed. State description: $(n; q_1, ..., q_i)$, where n is the number of busy servers and q_i the number of priority i customers in queue.
  - At time t, all queues are empty, n servers are busy: (n; 0, 0, 0, ...,0).
  - $r \equiv \max\{j, N_j \geq n\}$.
  - ${}_rX_{in}$ first passage time from state (n; 0, 0, 0, ...,0) to state $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$, i.e., to absorption in the subspace $(n-1; 0, ..., 0, q_r = 0, \bullet, \bullet, ..., \bullet)$.
  - a_k^c number of arrivals of priority k during ${}_rX_{in} + {}_{r+1}X_{in} + ... + {}_{k-1}X_{in}$, for $k \in \{i, ..., r+1\}$.
  - ${}_kX_{in}$ first passage time from state $(N_k; 0, ..., 0, q_k = a_k^c, \bullet, \bullet, ..., \bullet)$ to state $(N_k; 0, ..., 0, q_k = 0, \bullet, \bullet, ..., \bullet)$, for $k \in \{i, ..., r+1\}$.
  - $R_{i,n} \equiv {}_rX_{in} + {}_{r+1}X_{in} + ... + {}_iX_{in}$.

Table A.2 – Elementary Delay Cycles: Definition

REFERENCES

BENN, B.A. 1966

Hierarchical Car Pool Systems in Railroad Transportation, Ph.D. thesis, Case Institute of Technology, Cleveland, OH.

COBHAM, Alan. 1954.

Priority Assignment in Waiting Line Problems, *Opns. Res.* 2, pp.70-76.

COOPER, R.W. 1972.

Introduction to Queueing Theory, MacMillan, New York, NY.

COOPER, R.W. 1981.

Introduction to Queueing Theory, 2nd edition, North Holland, Elsevier, NY.

ERLANG, .1917.

GREEN, Linda. 1980.

A Queueing System in which Customers Require a Random Number of Servers, *Opns. Res.* 28, pp.1335-1346.

GREEN, Linda. 1984.

A Multiple Dispatch Queueing Model of Police Patrol Operations, *Mgt. Sci.* 30, pp.653-664.

GREEN, Linda and Peter Kolesar. 1983 (rev.12/84).

Testing the Validity of a Queueing Model of Police Patrol; Research Working Paper #521A, Columbia Business School, Columbia University, New York, NY.

JAISWAL, N.K. 1968.

Priority Queues, Acad. Press, New York, NY.

KLEINROCK, Leonard. 1975.

Queueing Systems, Vol.1 & 2. John Wiley & Sons, Inc., New York, N.Y.

REGE, Kiran M. and Bhaskar SENGUPTA. 1985.

A Priority Based Admission Scheme for a Multiclass Queueing System; *AT&T Technical Journal* 64, pp.1731-1753.

SCHAACK, Christian. 1985.

Cutoff Priority Queues: A Methodology for Police Patrol Dispatching; Massachusetts Institute of Technology, Operations Research Center, PhD thesis, December 1985.

SCHAACK, Christian and Richard C. LARSON. 1985.

An N Server Cutoff Multi-Priority Queue; Massachusetts Institute of Technology, Operations Research Center, Working Paper #OR 135-85.

TAYLOR, I.D.S. and J.G.C. TEMPLETON. 1980.

Waiting Time in a Multi-Server Cutoff-Priority Queue, and Its Application to an Urban Ambulance Service, *Opns. Res.* 28, pp.1168-1188.

WELCH, Peter D. 1964.

On a Generalized M/G1 Queueing Process in which the First Customer of Each Busy Period Receives Exceptional Service, *Opns.Res.* 12, pp.736-752.

WOLFF, R.W. 1982

Poisson Arrivals See Time Averages, *Opns.Res.* 30, pp.223-231.