MANAGEMENT SCIENCE
Vol. 30, No. 6, June 1984
Printed in U.S.A.

A MULTIPLE DISPATCH QUEUEING MODEL OF POLICE PATROL OPERATIONS*

LINDA GREEN
Graduate School of Business, Columbia University, New York, New York 10027

One of the primary concerns of urban police departments is the effective use of patrol cars. In large cities, police assigned to patrol cars typically account for more than 50% of total police manpower and their allocation has become particularly crucial in light of recent fiscal cutbacks.

The police patrol system is a complex multi-server queueing system, and recently many urban police departments have been using queueing models to estimate delays in responding to calls for police assistance. The magnitude of these delays is usually one basis for measuring system efficiency as well as for determining allocations of patrol cars among precincts. A major limitation of these models is that they assume that only a single unit is dispatched to each call. In general, this is not the case, particularly in police departments with one-officer patrol cars.

This paper describes a model that has been developed to represent patrol car operations more accurately. It is a multi-priority queueing model that explicitly reflects multiple car dispatches. Its purpose is not only to provide a better basis for the efficient allocation of patrol cars, but to enable police officials to gauge the effects of policies, such as one-officer patrol cars, which affect the number of cars dispatched to various types of incidents.

(QUEUES—MULTI-CHANNEL; QUEUES—PRIORITY; GOVERNMENT—SERVICES, POLICE)

1. Introduction

Typically, over half of the money and manpower of urban police departments goes into patrol car operations. The main functions of patrol cars are to respond quickly to emergencies and to maintain a "patrol presence" as a possible deterrent to crime.

The police patrol car system is a complex multi-server queueing system in which calls for service arrive randomly over time and require varying amounts of service time from one or more patrol cars. In recent years, many urban police departments in cities such as New York, Seattle, Los Angeles, Atlanta, Toledo, San Diego, and Minneapolis (Chaiken 1978) have been using queueing models to estimate dispatch queueing delay —the delay due to unavailability of patrol cars. The magnitude of these delays in each geographical command (precinct) is an important measure of the efficiency of the system and hence, along with other information, can be used to help determine allocations of cars. Several computer programs have been developed for specifying the number of patrol cars that should be assigned to each precinct of a city at various times of the day on each day of the week. A prominent example is the Patrol Car Allocation Model (PCAM), which was designed by Chaiken and Dormont (1978a, b) and which has been used by more than 40 police departments. More recently, the modelling of police patrol operations has become important in the evaluation of one- vs. two-officer patrol unit programs (Chelst 1981, Chelst and Barlach 1981).

The queueing models used to date have typically been Markovian multiple server models. PCAM, for example, uses an $M/M/c$ model with nonpreemptive priority classes originally developed by Cobham (1954). Precincts are defined as independently operating geographical commands, and key queueing statistics are calculated for each

*Accepted by Warren E. Walker; received April 22, 1982. This paper has been with the author 1 month for 2 revisions.

0025-1909/84/3006/0653$01.25

precinct for any given allocation of cars. Another approach has been to model the police patrol operation from the geographic perspective to obtain statistics on measures such as area-specific travel times and individual car workloads. The most general of these models is the hypercube queueing model developed by Larson (1974). This model explicitly represents individual servers and its major use has been to address problems related to the design of patrol areas and the location of emergency units. A bibliography of queueing (and other) models developed for the analysis of police deployment problems is given in Kolesar (1981).

One of the primary disadvantages of the queueing models currently in use is their inability to represent multiple car dispatches accurately. Every police department receives some calls that require the services of more than one patrol car. In New York City, for example, approximately 30% of all calls for service have a multiple car response, and at a considerable number of these, each car is busy for an extended period. This phenomenon is even more prevalent in systems with one officer in a patrol car. (In New York City there currently are two officers in a car.)

This paper describes a queueing model that has been developed to represent police patrol operations. In particular, it is a multi-server, multi-priority queueing model in which the number of servers assigned to each customer is a random variable dependent on the customer's "type" and the availability of servers. This model is based on a simpler queueing model developed by Green (1980). It has been used in a study of one- vs. two-officer patrol car programs in New York City (see Green and Kolesar 1982) and has been incorporated in a modernized version of PCAM. Comparisons of queueing statistics show that this model is significantly more accurate than the Cobham model it replaces (see Green and Kolesar 1984, 1983).

A description of the basic model and some preliminary results are given in §§2 and 3. §4 contains the derivations of several measures of performance under the assumption that the number of servers used by a customer is independent of system congestion. These include the probability of delay and the expected delay for each priority class, and the average overall utilization of servers. §5 describes the extension of this model to allow for the number of servers assigned to a customer to depend on server availability and §6 briefly discusses the model's uses.

2. Model Description and Definitions

We consider a queueing system with s identical servers. Jobs (calls for service) arrive according to a Poisson process with rate λ. With probability p_k the call belongs to priority class k, $1 \leqslant k \leqslant n$. Jobs of priority k are selected for service ahead of jobs of priority l, $l > k$, on a first-come-first-served basis within that class. In the basic model (details are in §4), a job in class k requires service from i servers with probability $c_k(i)$, $1 \leqslant i \leqslant s$, independent of the state of the system. In the more general model of §5, a job in class k requires a minimum of i servers and a maximum of j servers with probability $c_k(i, j)$, $1 \leqslant i \leqslant j \leqslant s$. The actual number used is dependent on the availability of servers at the job's arrival epoch. In both models, a job that arrives at an empty queue begins service immediately if at least the (minimum) number of servers required is available. Otherwise when a job becomes first in queue, servers are "assigned" to it as they free up and service begins only when the (minimum) number required are available. (Thus, jobs and servers can both be waiting simultaneously.) When service begins, each server's "completion time" has a distribution that is exponential with mean $1/\mu$. Once a job is first in queue and has been assigned at least one server, it cannot be preceded by a higher priority customer. This logic resembles actual police dispatch operations that we have observed.

Individual server completion times are assumed to be statistically independent once work is begun, and so servers associated with the same job free up one at a time. This assumption is empirically supported as a good representation of actual patrol car behavior. Analyses performed on data from New York City show that the correlation of service times of cars working on the same job are low, and that cars leave the incident individually in a pattern that resembles that predicted by the assumption of independent, exponential service times. A detailed analysis of the model's overall correspondence to actual operations is given in Green and Kolesar (1983).

This model is an extension of an earlier queueing model presented in Green (1980), which assumes no priorities and that each job is assigned a fixed number of servers independent of the state of the system. The major result presented in Green (1980) is an expression for the distribution of time until a job begins service (defined below as full delay). In this paper, we obtain several measures of performance for the priority system which are analogous to results in Green (1980) and we draw upon those results in the derivations. However, the focus of this paper is the direct derivation for each priority class of the expectations of the various delays that are used to measure performance in police patrol applications.

Before proceeding with the analysis, we define some random variables. For easy reference, Table 1 lists the symbols and definitions used throughout the paper. The *queue period*, Q, is the peiod of time beginning when a job arrives to an empty queue but must wait for service, and ending when the queue next becomes empty. Similarly, the *nonqueue period*, $\bar{Q}$, begins when the preceding queue period ends and lasts until a queue next forms. B_k, the *interservice time* of a class k customer, is defined only for jobs that arrive during a queue period, and is the total time spent in the first queue

TABLE 1

List of Symbols and Definitions

Symbol	Definition
p_k	Probability of class k job
$c_k(i)$	Probability that class k job requires i servers (basic model)
$c_k(i, j)$	Probability that class k job requires a minimum of i and a maximum of j servers (state-dependent dispatch)
Q	Length of queue period
$\bar{Q}$	Length of nonqueue period
B_k	Time spent in 1st queue position by class k job that arrives during queue period
D_k	Time spent in 1st queue position by class k job that initiates queue
$W_I(k)$	Delay until 1st server assigned for class k job
$W_F(k)$	Delay until all servers assigned for class k job
$W_S(k)$	Staging delay for class k job = $W_F(k) - W_I(k)$
p_q	Probability of queue
q_i	Probability of i busy servers during queue period
$\bar{q}_i$	Probability of i busy servers during nonqueue period
$p_d(k)$	Probability that class k arrival to nonqueue period is delayed
RB_k	Equilibrium excess time in 1st position of class k job
λ_k	Class k arrival rate
$p_q(k)$	Probability that class k job is first in queue
$cd_k(i)$	Probability that a class k job is served by i servers (state-dependent dispatch)

position prior to entering service. (In the nonpriority case, the interservice time is the time between service start epochs when there is a queue.) Similarly, the *queue initiator's delay*, D_k, of a class k job is the time spent in the first queue position prior to entering service by a class k job that initiates a queue period. So D_k will be this job's queueing delay if a higher priority arrival does not precede it into service.

The *initial delay*, $W_I(k)$, for a class k job is the interval between its arrival epoch and the time at which the first server is assigned. The waiting time in queue or *full delay*, $W_F(k)$, is the delay experienced by a class k job until all servers to be assigned to it are available and service begins. The difference between the full delay and the initial delay is called the *staging delay*, $W_S(k)$. This delay is useful in assessing police officer safety (particularly in one-officer programs) since it measures one component of the time during which the first car dispatched waits for backup units. (The other component involves the difference in travel times (Chelst 1981, Chelst and Barlach 1981).)

3. Preliminary Results

The analyses which appear in the following sections will use the results presented below. Many of these results are analogous to those in Green (1980 or 1981) and will therefore be presented without detailed proofs.

PROPOSITION 1. *All s servers are busy immediately after a customer enters service during a queue period. (Note that a server is considered "busy" only if he is actively serving a customer.)*

PROOF. The proof is identical to that of Proposition 1 in Green (1980) once the observation is made that every job that starts service during a queue period spends a positive amount of time in queue. This is due to the assumption that a higher priority job cannot precede the first job in queue into service if at least one server has already been assigned to that job.

Define the sum of the nonqueue period $\overline{Q}$ and the queue period Q to be the queueing cycle. Since by Proposition 1, all s servers are busy at the start of each nonqueue period, we have the following result.

COROLLARY 1. *The sequence of queueing cycles forms a renewal process and the queue-length process is regenerative.*

Let p_q be the steady-state probability that there exists a queue. From Corollary 1,

$$p_q = E(Q)/\left[E(\overline{Q}) + E(Q)\right]. \tag{1}$$

It can be shown (see Green 1981, Theorem 6) that $E(Q)$ and $E(\overline{Q})$ are identical to the corresponding quantities in the nonpriority version of the model. Therefore from Green (1980) we have

$$E(Q) = E(D)/\left[1 - \lambda E(B)\right] \qquad \text{where} \tag{2}$$

$$E(D) = \sum_{k=1}^{n} p_k E(D_k) \quad \text{and} \quad E(B) = \sum_{k=1}^{n} p_k E(B_k).$$

$E(B_k)$ and $E(D_k)$ are derived below. $E(\overline{Q})$ is obtained by considering the process

$$\overline{N}_\beta(t) = \begin{cases} \text{number of busy servers at } t & \text{if no queue at } t, \\ s+1 & \text{otherwise,} \end{cases} \tag{3}$$

where $t = 0$ marks the beginning of an arbitrary nonqueue period. Let T be the transition matrix of transient states for the Markov chain embedded in $\{\bar{N}_\beta(t),\ t \geqslant 0\}$ at arrival and server completion epochs and with absorbing state $s+1$. This matrix T will differ for the two models presented in this paper. For the basic model of §4, it is given by

$$T = \begin{array}{c|ccccc|}
 & 0 & 1 & 2 & \cdots & s \\
0 & 0 & c(1) & c(2) & & c(s) \\
1 & \mu/(\lambda+\mu) & 0 & c(1)\lambda/(\lambda+\mu) & & c(s-1)\lambda/(\lambda+\mu) \\
2 & 0 & 2\mu/(\lambda+2\mu) & 0 & & c(s-2)\lambda/(\lambda+2\mu) \\
\vdots & \vdots & & & & \vdots \\
s & 0 & & & & 0
\end{array} \tag{4}$$

where $c(i) = \sum_{k=1}^{n} p_k c_k(i)$. In §5, T will be determined for the general model. From Green (1980),

$$E(\bar{Q}) = \sum_{j=0}^{s} V_{sj}/(\lambda + j\mu) \qquad \text{where} \tag{5}$$

$$V = (I - T)^{-1} \tag{6}$$

is the fundamental matrix of the Markov chain with entries V_{ij} = the mean number of visits to state j starting in state i (see Kemeny and Snell 1960, Chapter 3).

We will also need the distribution of busy servers during a nonqueue period, $\{\bar{q}_i\}$, and during a queue period, $\{q_i\}$. These distributions are identical to those for the nonpriority system (see Green 1981, Theorem 3). From Green (1980)

$$\bar{q}_i = V_{si}/(\lambda + i\mu)E(\bar{Q}), \qquad i = 0, 1, \ldots, s, \quad \text{and} \tag{7}$$

$$q_i = \frac{\lambda E(Q)\sum_{k=s-i+1}^{s} c(k) + \sum_{j=i}^{s} \bar{q}_j \sum_{k=s-i+1}^{s} c(k)}{i\mu E(Q)\sum_{k=1}^{s} \bar{q}_k \sum_{j=1}^{i} c(s-i+j)}, \qquad i = 1, \ldots, s. \tag{8}$$

We now derive the expected values for B_k and D_k. Recall that B_k is the total amount of time spent in the first queue position by a priority k customer who arrives to a queue. This consists of two components—the time until the first server is assigned and the interval of the time that begins when the first server is assigned and ends when all of the necessary servers have been assigned. Since a job cannot be bumped out of first position after the first server is assigned, and since by Proposition 1 all servers are busy when an interservice time begins, the expected value of the second component for a customer requiring i servers is clearly $1/(s-1)\mu + 1/(s-2)\mu + \cdots + 1/(s-i+1)\mu$. To obtain the expected value of the first component, $E(X_k)$, we condition on whether or not the job is bumped out of first position. We get

$$\begin{aligned}
E(X_k) &= E(X_k \mid \text{job not bumped})\Pr(\text{not bumped}) \\
&\quad + E(X_k \mid \text{job bumped})\Pr(\text{bumped}) \\
&= \left(\frac{1}{\sum_{i=1}^{k-1}\lambda_i + s\mu}\right) \cdot \left(\frac{s\mu}{\sum_{i-1}^{k-1}\lambda_i + s\mu}\right) \\
&\quad + \left(\frac{1}{\sum_{i=1}^{k-1}\lambda_i + s\mu} + E(X_k)\right)\left(\frac{\sum_{i=1}^{k-1}\lambda_i}{\sum_{i=1}^{k-1}\lambda_i + s\mu}\right).
\end{aligned}$$

Solving for $E(X_k)$ yields $E(X_k) = 1/s\mu$ and so

$$E(B_k) = \sum_{i=1}^{s} \left[1/s\mu + 1/(s-1)\mu + \cdots + 1/(s-i+1)\mu\right] c_k(i), \qquad k = 1, \ldots, n. \tag{9}$$

To obtain $E(D_k)$, we must consider the number of busy servers found by the initiating job of the queue period as well as the number of servers it requires. (Note that when all servers are busy, $E(D_k) = E(B_k)$ by the argument given above.) Define $H_k(i, j)$ to be the joint probability that a job of class k finds i busy servers at its arrival epoch and needs j servers given that it initiates a queue period. From the assumption of Poisson arrivals,

$$H_k(i, j) = \begin{cases} \bar{q}_i c_j / p_d(k), & j > s - i, \\ 0, & j \leqslant s - i, \end{cases} \qquad k = 1, \ldots, n, \tag{10}$$

where $p_d(k)$ is the probability that a class k arrival to a nonqueue period has a positive delay and is given by

$$p_d(k) = \sum_{i=1}^{s} \bar{q}_i \sum_{j=1}^{i} c_k(s-i+j), \qquad k = 1, \ldots, n. \tag{11}$$

Therefore

$$E(D_k) = \sum_{i=1}^{s} \sum_{j=1}^{i} \left[1/i\mu + 1/((i-1)\mu) + \cdots + 1/((i-j+1)\mu)\right] \bar{q}_i c_k(s-i+j)/p_d(k), \qquad k = 1, \ldots, n. \tag{12}$$

We will also need $E(RB)_k$, the equilibrium expectation of the remaining time in first queue position of a class k job. Reasoning as above, we get

$$E(RB_k) = \sum_{i=1}^{s} \sum_{j=1}^{i} \left[1/i\mu + 1/(i-1)\mu + \cdots + 1/(i-j+1)\mu\right] \frac{q_i c_k(s-i+j)}{\sum_{j=s-i+1}^{s} c_k(j)}, \qquad k = 1, \ldots, n. \tag{13}$$

4. Basic Priority Model

We assume that jobs in priority class k arrive according to a Poisson process at rate $\lambda_k = p_k \lambda$ and receive service from i servers with probability $c_k(i)$, $1 \leqslant i \leqslant s$, regardless of the level of system congestion. We will first present results for two measures of performance that are of particular interest in police patrol operations: $E(NP)$, the average number of available servers (not servicing or assigned to a job); and pdel$_k$, the fraction of priority k calls that have a positive delay.

PROPOSITION 2. *$E(NP) = (1 - p_q)\sum_{i=0}^{s} i\bar{q}_{s-i}$where $\bar{q}_i$ is given by* (7) *and p_q by* (1).

PROOF. Cars are available only during a nonqueue period. Since the probability that i servers are available during a nonqueue period is $\bar{q}_{s-i}$, we get the result.

PROPOSITION 3. *$pdel_k = p_q + (1 - p_q)p_d(k)$, $k = 1, \ldots, n$, where $p_d(k)$ is given by* (11).

PROOF. The result follows directly from the definitions of p_q and $p_d(k)$.

We now derive the expected values for the initial delay and the full delay in queue by priority class. This will be done by exploiting the $M/G/1$ structure embedded in the system (see Green 1980) and using an approach based on Cobham (1954) for the $M/G/1$ queue with nonpreemptive priorities.

In the following analysis, we assume that the system is in steady-state. Green (1980), a sufficient condition for the existence of a steady-state solution in the nonpriority system is given by $\lambda E(B) < 1$. It can easily be shown that this condition is also sufficient in the priority case. Since $\lambda E(B) = \sum_{k=1}^{n} \lambda_k E(B_k)$, this condition implies that $\lambda_k E(B_k) < 1$, $k = 1, \ldots, n$.

In general, a job's full delay in queue will be the sum of three components: (1) the total time spent in the first queue position by those jobs of equal or higher priority in queue at its arrival epoch; (2) the total time spent in the first queue position by those jobs of higher priority which arrive before this job is assigned a server; and (3) its own time spent as first in queue.

Consider an arbitrary class 1 arrival. Since it has priority over jobs of any other class, its waiting time will be affected only by the priority 1 jobs in queue at its arrival epoch and by the existence of a lower priority job in the first queue position that has had at least one server assigned. To calculate $E(W_F(1))$ we condition on L_1, the number of class 1 jobs that the arrival finds in queue, and K, the class of the first job in queue, if any. Define $K = 0$ when there is no queue.

First consider the case when this job arrives to an empty queue. Its waiting time is just its own delay in entering service, D_1, if it is delayed, and zero otherwise. So

$$E(W_F(1) \mid K = 0) = E(D_1) p_d(1) \tag{14}$$

where $E(D_1)$ is given by (12) and $p_d(1)$ by (11). Now suppose that $K = 1$ and $L_1 = n_1$, $n_1 \geqslant 1$. The waiting time of the class 1 arrival will be the sum of the remaining interservice time of the first job in queue, RB_1, the interservice times of the other $n_1 - 1$ class 1 jobs in queue, and its own interservice time. So

$$E(W_F(1) \mid L_1 = n_1, K = 1) = E(RB_1) + (n_1 - 1)E(B_1) + E(B_1), \qquad n_1 \geqslant 1, \tag{15}$$

where $E(B_1)$ is given by (9) and $E(RB_1)$ by (13).

Finally, if $L_1 = n_1$, $n_1 \geqslant 0$ and the first job in queue is of a lower priority class i, $i \geqslant 2$, the waiting time of the class 1 arrival will be the sum of the interservice times of the n_1 class 1 jobs in queue, its own interservice time, and the remaining interservice time, RB_i, of the first job in queue if at least one server has been assigned to it. Note that the probability that at least one server has been assigned is $1 - q_s$ since a server does not become busy until all required servers are available to begin service. So

$$E(W_F(1) \mid L_1 = n_1, K = i) = n_1 E(B_1) + E(B_1) + E(RB_i)(1 - q_s), \qquad n_1 \geqslant 0, \quad i \geqslant 2. \tag{16}$$

Since the distribution of the interservice time is independent of the number of jobs in queue, (14), (15), and (16) yield

$$E(W_F(1)) = E(L_1)E(B_1) + E(D_1)p_d(1)(1 - p_q) + E(RB_1)p_q(1) + \sum_{i=2}^{n} \left[E(B_1) + E(RB_i)(1 - q_s) \right] p_q(i) \tag{17}$$

where $p_q(i)$ is the steady-state probability that a queue exists and that the first job in queue is of class i. An expression for $p_q(i)$ will be derived later. Using Little's formula and solving for $E(W_F(1))$ we obtain

$$E(W_F(1)) = \frac{E(R_1)}{1 - \lambda_1 E(B_1)} \qquad \text{where} \tag{18}$$

$$E(R_1) = E(D_1)p_d(1)(1 - p_q) + E(RB_1)p_q(1) + \sum_{i=2}^{n} \left[E(B_1) + E(RB_i)(1 - q_s) \right] p_q(i). \tag{19}$$

The expected initial delay of a class 1 job can be determined directly as above. However, it will be more convenient to express it in terms of the class 1 expected waiting time. Recall that the initial delay is the delay until a job is assigned one server. Therefore, if a class 1 job arrives at a queue and needs, for example, j servers, its initial delay will be its full delay less the time it takes for the remaining $j - 1$ required servers to free up. Since $E(B_1)$ is the expected wait until a class 1 job is assigned all the servers it needs once it is first in queue, and $1/s\mu$ is the mean time until the first server frees,

$$E(W_I(1) \mid \text{queue}) = E(W_F(1) \mid \text{queue}) - \left[E(B_1) - 1/s\mu \right]. \tag{20}$$

If the class 1 job arrives to an empty queue, its initial delay will be zero if at least one server is free. If all servers are busy, its expected initial delay will be $1/s\mu$. Therefore

$$E(W_I(1) \mid \text{no queue}) = \bar{q}_s / s\mu. \tag{21}$$

Since $E(W_F(1) \mid \text{no queue}) = E(D_1)p_d(1)$, (21) can be rewritten as

$$E(W_I(1) \mid \text{no queue}) = E(W_F(1) \mid \text{no queue}) - \left[E(D_1)p_d(1) - \bar{q}_s / s\mu \right] \tag{22}$$

and, therefore,

$$E(W_I(1)) = E(W_F(1)) - E(W_S(1)) \tag{23}$$

where $E(W_S(1))$ is the mean staging delay of a class 1 job and is given by

$$E(W_S(1)) = \left[E(B_1) - 1/s\mu \right] p_q + \left[E(D_1)p_d(1) - \bar{q}_s / s\mu \right](1 - p_q). \tag{24}$$

Note that this reasoning applies to jobs of any priority class. So in general we have

$$E(W_I(k)) = E(W_F(k)) - E(W_S(k)), \qquad k = 1, \ldots, n, \qquad \text{where} \tag{25}$$

$$E(W_S(k)) = \left[E(B_k) - 1/s\mu \right] p_q + \left[E(D_k)p_d(k) - \bar{q}_s / s\mu \right](1 - p_q), \qquad k = 1, \ldots, n. \tag{26}$$

To calculate the expected full delay of a class 2 job, we must condition on the number of class 1 (L_1) and class 2 jobs in queue (L_2) at its arrival epoch, the identity of the first job in queue (K), and the number of class 1 jobs that arrive subsequent to its arrival epoch and precede it into service ($L^{(1)}$). First consider a class 2 job that arrives to an empty queue. Its delay is the time it spends waiting in first queue position, if any, plus the sum of the interservice times of the $L^{(1)}$ class 1 customers who arrive before the first server is assigned. ($L^{(1)}$ can be positive, of course, only if all servers are busy at this job's arrival epoch.) So

$$E\left(W_F(2) \mid L^{(1)} = n^{(1)}, K = 0\right) = E(D_2)p_d(2) + n^{(1)}E(B_1), \qquad n^{(1)} \geqslant 0. \tag{27}$$

Now suppose that $K = 1$, $L_1 = n_1 \geqslant 1$, $L_2 = n_2 \geqslant 0$, and $L^{(1)} = n^{(1)} \geqslant 0$. The class 2 arrival will have to wait for the remaining interservice time of the first class 1 job, the $n_1 - 1$ other class 1 interservice times, the interservice times of the n_2 class 2 jobs, the

$n^{(1)}$ interservice times of the class 1 arrivals, and its own interservice time. So

$$E\big(W_F(2)\,|\,L_1 = n_1, L_2 = n_2, L^{(1)} = n^{(1)}, K = 1\big)$$
$$= E(RB_1) + (n_1 - 1)E(B_1) + n_2E(B_2) + n^{(1)}E(B_1) + E(B_2),$$
$$n_1 \geqslant 1, \quad n_2 \geqslant 0, \quad n^{(1)} \geqslant 0. \quad (28)$$

If $K = 2$, $L_1 = n_1$, $L_2 = n_2$, and $L^{(1)} = n^{(1)}$, the class 2's delay is the sum of the remaining interservice time of the first class 2 job, the interservice times of the other $n_2 - 1$ class 2 jobs, the n_1 interservice times of the class 1 jobs in queue, and $n^{(1)}$ interservice times of the subsequent class 1 arrivals and its own interservice time. This gives

$$E\big(W_F(2)\,|\,L_1 = n_1, L_2 = n_2, L^{(1)} = n^{(1)}, K = 2\big)$$
$$= E(RB_2) + (n_2 - 1)E(B_2) + n_1E(B_1) + n^{(1)}E(B_1) + E(B_2),$$
$$n_1 \geqslant 0, \quad n_2 \geqslant 0, \quad n^{(1)} \geqslant 0. \quad (29)$$

For the case when $K = i > 2$, the remaining interservice time of the first customer in queue is added to the class 2 arrival's waiting time only if at least one server is free. Therefore,

$$E\big(W_F(2)\,|\,L_1 = n_1, L_2 = n_2, L^{(1)} = n^{(1)}, K = i\big)$$
$$= n_1E(B_1) + n_2E(B_2) + n^{(1)}E(B_1) + E(B_2) + E(RB_i)(1 - q_s),$$
$$n_1 \geqslant 0, \quad n_2 \geqslant 0, \quad n^{(1)1} \geqslant 0, \quad i > 2. \quad (30)$$

From (28), (29), and (30) we obtain

$$E(W_F(2)) = E(L_1)E(B_1) + E(L_2)E(B_2) + E(L^{(1)})E(B_1) + E(D_2)p_d(2)(1 - p_q)$$
$$+ \big[E(RB_1) - E(B_1) + E(B_2)\big]p_q(1) + E(RB_2)p_q(2)$$
$$+ \sum_{i=3}^{n}\big[E(B_2) + E(RB_i)(1 - q_s)\big]p_q(i). \quad (31)$$

Note that the interval of time during which class 1 jobs that arrive subsequent to a class 2's arrival epoch will precede it into service is exactly the class 2's initial delay $W_I(2)$. So from the assumption of Poisson arrivals,

$$E(L^{(1)}) = \lambda_1 E(W_I(2)). \quad (32)$$

Using Little's formula, (25) and (32), we can solve for $E(W_F(2))$ to get

$$E(W_F(2)) = \big[\lambda_1 E(W_F(1))E(B_1) + E(R_2)\big]/\big[1 - \lambda_1 E(B_1) - \lambda_2 E(B_2)\big] \quad \text{where} \quad (33)$$

$$E(R_2) = E(D_2)p_d(2)(1 - p_q) + \big[E(RB_1) - E(B_1) + E(B_2)\big]p_q(1)$$
$$+ E(RB_2)p_q(2) + \sum_{i=3}^{n}\big[E(B_2) + E(RB_i)(1 - q_s)\big]p_q(i)$$
$$- \lambda_1 E(B_1)E(W_S(2)). \quad (34)$$

Reasoning as above, it can be easily shown that for an arbitrary job of class k, the expected waiting time in queue is given by

THEOREM 1.

$$E(W_F(k)) = \frac{\sum_{i<k}\lambda_i E(W_F(i))E(B_i) + E(R_k)}{1 - \sum_{i\leqslant k}\lambda_i E(B_i)} \qquad \textit{where}$$

$$\begin{aligned} E(R_k) = {} & E(D_k)p_d(k)(1-p_q) + \sum_{i<k}\left[E(RB_i) - E(B_i) + E(B_k)\right]p_q(i) \\ & + E(RB_k)p_q(k) + \sum_{i>k}\left[E(B_k) + E(RB_i)(1-q_s)\right]p_q(i) \\ & - \sum_{i<k}\lambda_i E(B_i)E(W_S(k)). \end{aligned}$$

To obtain $p_q(k)$, $k = 1, \ldots, n$, we apply Little's formula to the "system" defined by the first queue position. A class k job will (eventually) occupy the first queue position under two possible conditions: it arrives during a queue period, or it arrives to an empty queue but has a positive delay. In the first case, its arrival rate to the first queue position is $\lambda_k p_q$ and its expected "waiting time" there is $E(B_k)$. (As explained previously, this is true even though it may be bumped out of first position.) In the second case, its arrival rate to first position is $\lambda_k(1-p_q)p_d(k)$ and its expected wait there is $E(D_k)$. Since $p_q(k)$ is equivalent to the expected number of class k jobs in first queue position in steady state, Little's formula results in

$$p_q(k) = \lambda_k p_q E(B_k) + \lambda_k(1-p_q)p_d(k)E(D_k), \qquad k = 1, \ldots, n. \tag{35}$$

5. Model with State-Dependent Dispatch

In this section, we extend the model of §4 to allow for the number of servers assigned to a job to depend on server availability. Specifically, each priority class k has an associated joint probability distribution on the minimum and maximum number of servers required by a job in that class. The dispatch protocol is as follows: A job that joins a queue upon its arrival will, when it reaches first position, start service when the minimum number of servers needed is available. If it arrives at an empty queue and the number of available servers is less than the minimum number required, it will again be serviced by this minimum number of servers when they become available. However, if there is no queue at its arrival epoch and the number of available servers is at least the minimum required, it will start service immediately with all the servers available up to the maximum number required.

Define $c_k(i) = \sum_{j=i}^{s} c_k(i, j)$, $i = 1, \ldots, s$, to be the probability that the minimum number of servers needed by a class k job is i. Since a job will always be assigned the minimum number of servers required unless it arrives during a nonqueue period and more than this number is available, the expressions derived in §3 for $E(B_k)$, $E(D_k)$, pd_k, $E(RB_k)$ and $E(Q)$ are all valid for this model as well as for the model of §4. In fact, the only difference between the two systems is in the number of servers that become busy during a nonqueue period. In particular, all of the results of the previous sections hold for this "flexible" dispatch model except for the matrix T given by (4) in §3.

Recall that T is the transition matrix of transient states for the Markov chain embedded in the process $\{\overline{N}_\beta(t), t \geqslant 0\}$ defined by (3) and with absorbing state $s + 1$. As before, the transition probabilities corresponding to service completions are given by

$$T_{i,i-1} = i\mu/(\lambda + i\mu), \qquad i = 1, \ldots, s. \tag{36}$$

However, the probability of a transition from i to j, $i < j < s$, is now governed by the distribution of the maximum number of servers needed. Let $cm_k(j) = \sum_{i=1}^{j} c_k(i, j)$ be the probability that the maximum number of servers required by a class k job is j, and let $cm(j) = \sum_{k=1}^{n} p_k cm_k(j)$. Then

$$\begin{aligned} T_{0,j} &= cm(j), && j = 1, \dots, s-1, \\ T_{i,i+j} &= cm(j)\lambda/(\lambda + i\mu), && i = 1, \dots, s-2, \quad j = 1, \dots, s-i-1. \end{aligned} \tag{37}$$

Transitions from i to s will occur only if a job arrives when there are i servers busy and requires a minimum of $s - i$ or fewer servers (and so can enter service immediately) and a maximum of $s - i$ or more servers (so that it is assigned all servers available causing all s to be busy). Let $c(i, j) = \sum_{k=1}^{n} p_k c_k(i, j)$. Then

$$T_{0,s} = cm(s), \qquad T_{i,s} = \sum_{j=1}^{s-i} \sum_{l=s-i}^{s} c(j, l), \qquad i = 1, \dots, s-1. \tag{38}$$

All other entries of T will be zero. So by using the T matrix defined by (36), (37), and (38) in equation (6) of §3, all of the delay and server utilization results of §4 can now be applied to this state-dependent dispatch system.

One of the interesting consequences of using a state-dependent dispatch protocol is that as the traffic intensity increases, the magnitude of delays will not increase as dramatically as in a state-independent dispatch system. This is because the system compensates to some extent by assigning fewer servers to customers when fewer servers are available. So, though system performance as measured, e.g., by probability of delay may not deteriorate significantly for a given increase in the arrival rate or a decrease in the number of servers in the system, the level of service will decrease with respect to the number of servers assigned per job. Therefore, it is of interest to obtain the distribution of the number of servers actually assigned per job for each priority class.

This can be done by conditioning on the minimum and maximum number of servers required by a job, and the number of servers busy at an arrival epoch. The resulting probability $cd_k(i)$ that a class k job is served by i servers is given by

$$\begin{aligned} cd_k(i) = c_k(i) p_q + \Bigg[& c_k(i) \sum_{j=1}^{i} \bar{q}_{s-j} + c_k(i, i) \sum_{j=i+1}^{s} \bar{q}_{s-j} \\ & + \bar{q}_{s-i} \sum_{m<i} \sum_{n>i} c(m, n) + \sum_{m<i} c(m, i) \sum_{j=i}^{s} \bar{q}_{s-j} \Bigg] (1 - p_q), \\ & \qquad k = 1, \dots, n, \quad i = 1, \dots, s. \end{aligned} \tag{39}$$

6. Applications

The multiple-dispatch queueing model was originally developed in response to a request from the New York City Police Department to study apparent underestimates of dispatch queue delays produced by its implementation of PCAM. There was particular concern regarding PCAM's inability to explicitly represent the existence of multiple car dispatches, which account for about 30% of all calls for service in New York. PCAM attempts to represent this phenomenon by increasing the arrival rate in the Cobham model by a factor equal to the average number of cars dispatched. For example, if a precinct has a call rate of 4 calls per hour and the average number of responding cars per call is 1.4, the adjusted call rate would be 5.6. In Green and Kolesar (1984b) we show that this type of approximation always results in lower predictions of delay than those produced by the multiple-dispatch model, and often substantially so. Comparison of queueing statistics from both models with empirical data from New York City indicates that the multiple-dispatch model is consistently more accurate in predicting actual delays (see Green and Kolesar 1983).

In addition to its use as an aid in determining allocations of police patrol cars, the model has proven to be a valuable tool in evaluating policies which affect the number of cars dispatched to various incidents. This was demonstrated in a feasibility study of a one-officer car program in New York City (see Green and Kolesar 1984a) in which the model presented in this paper was the major tool for analysis. Though it was clear that more one-officer cars would be needed at many types of incidents to provide adequate police manpower, it was unknown how many additional one-officer cars would be needed citywide to achieve an equivalent level of performance as with two-officer cars. The multiple-dispatch queueing model was well suited to the fundamental purpose of the study—analyzing the effect of differing numbers of cars per emergency call under one- or two-officer patrol operations. It also had the advantage of allowing for the consideration of the full delay and staging delay in the comparisons of alternatives, as well as the initial delay—the measure produced by previous queueing models. These delays are important measures in the evaluation of a one-officer program because to insure police officer safety an adequate number of cars must respond quickly to potentially dangerous incidents.

In conclusion, the multiple-dispatch queueing model described in this paper appears to be a more complete model of police patrol operations for cities in which multiple car responses are prevalent. Its demonstrated usefulness as an aid in decisionmaking should make it a valuable tool for urban police departments.[1]

[1] This work was supported by a grant from the National Institute of Justice and by the Faculty Research Fund of the Graduate School of Business, Columbia University.

References

CHAIKEN, J. M., "Transfer of Emergency Service Deployment Models to Operating Agencies," *Management Sci.*, 24 (1978), 719–731.

——— AND P. DORMONT, "A Patrol Car Allocation Model: Background," *Management Sci.*, 24 (1978a), 1280–1290.

——— AND ———, "A Patrol Car Allocation Model: Capabilities and Algorithms," *Management Sci.*, 24 (1978b), 1291–1300.

CHELST, K. R., "Deployment of One- vs. Two-Officer Patrol Units: A Comparison of Travel Times," *Management Sci.*, 27 (1981), 213–230.

——— AND Z. BARLACH, "Multiple Unit Dispatches in Emergency Services," *Management Sci.*, 27 (1981), 1390–1409.

COBHAM, A., "Priority Assignment in Waiting Line Problems," *Oper. Res.*, 2 (1954), 70–76.

GREEN, L., "A Queueing System in which Customers Require a Random Number of Servers," *Oper. Res.*, 28 (1980), 1335–1346.

———, "Comparing Operating Characteristics of Queues in which Customers Require a Random Number of Servers," *Management Sci.*, 27 (1981), 65–74.

——— AND P. KOLESAR, "The Feasibility of One-Officer Patrol Cars in New York City," *Management Sci.*, 30 (1984a), forthcoming.

——— AND ———, "A Comparison of the Multiple Dispatch and $M/M/c$ Models of Police Patrol," *Management Sci.*, 30 (1984b), 665–670.

——— AND ———, "Testing the Validity of a Queueing Model of Police Patrol," Research Working paper, No. 521A, Graduate School of Business, Columbia University, 1983.

KEMENY, J. G. AND J. C. SNELL, *Finite Markov Chains*, D. Van Nostrand, Princeton, N.J., 1960.

KOLESAR, P., "Ten Years of Research on the Logistics of Urban Emergency Services," *Operational Research '81*, J. P. Braus (Ed.), North-Holland, Amsterdam, 1981, 557–568.

LARSON, R. C., "A Hypercube Queueing Model for Facility Location and Redistricting in Urban Emergency Services," *Computers and Oper. Res.*, 1 (1974), 67–95.