Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

THE IMPACT OF PRIORITY GENERATIONS IN A MULTI-PRIORITY QUEUEING SYSTEM - A SIMULATION APPROACH

A. Krishnamoorthy

Department of Mathematics
Cochin University of Science and Technology
Kochi 682022, India

Viswanath C. Narayanan

Department of Mathematics
Government Engineering College
Sreekrishnapuram, 679 513, India

Srinivas R. Chakravarthy

Department of Industrial and Manufacturing Engineering
Kettering University, Flint, MI 48504, USA.

ABSTRACT

In this paper, we consider a preemptive (multiple) priority queueing model in which arrivals occur according to a Markovian arrival process (MAP). An arriving customer belongs to priority type i, $1 \leq i \leq m+1$, with probability p_i. The highest priority, labeled as 0, is generated by other priority customers while waiting in the system and not otherwise. Also, a customer of priority i can turn into a priority $j, j \neq i, 1 \leq i, j \leq m+1$, customer, after a random amount of time that is assumed to be exponentially distributed with parameter depending on the priority type. The waiting spaces for all but priority type $m+1$ are assumed to be finite. The $(m+1)-st$ priority customers have unlimited waiting space. At any given time, the system can have at most one highest priority customer. Thus, all priority customers except the $(m+1)-st$ are subject to loss. Customers are served on a first-come-first-served basis within their priority by a single server and the service times are assumed to follow a phase type distribution that may depend on the customer priority type. This queueing model, which is a level-dependent quasi-birth-and-death process, is amenable for investigation algorithmically through the well-known matrix-analytic methodology. However, here we propose to study through simulation using ARENA, a powerful simulation software as some key measures such as the waiting time distributions are highly complex to characterize analytically. The simulated results for a few scenarios are presented.

1 INTRODUCTION

While priority queues (both pre-emptive and non pre-emptive) have been extensively studied (see Takagi (Takagi 1989), Jaiswal (Jaiswal 1968) and references therein) in the literature, in this paper we analyze a multi-priority queueing system attended by a single server in which arriving priority customers can change their priorities while waiting for service. Such priority queueing models have been motivated by applications in areas such as health care and communication, and are referred to as self-generated priority queues. Self generation of priorities is a common phenomenon. For example, patients waiting in a clinic can become seriously ill and therefore given preference over other waiting patients. The change of priorities also takes place in multi-speciality hospitals. For example a patient waiting for an appointment with a physician specialized in a certain ailment, may require other specialized treatments. This patient may turn out to be critically ill at which time becomes a super-priority type requiring immediate attention in the current place (pre-empting the other priority customer in service) or elsewhere (due to another highest priority in service) by leaving the system. One can also find applications of the present model in communication, aircraft landing and so on.

Krishnamoorthy, Deepak and Viswanath (Krishnamoorthy, Deepak, and Narayanan 2002) introduced self-generation of priority into queueing models. Wang (Wang 2004) discusses patient queue models with self-generation of priorities (though he does not introduce that terminology). He assumes all time variables to be exponentially distributed. Krishnamoorthy, Viswanath and Deepak (Krishnamoorthy, Narayanan, and Deepak 2005) provide an extensive analysis of a multiserver queue with Poisson arrival process. Customers while joining the system, are categorized as being ordinary; however while waiting they generate into priority according to a linear rate. The system is shown to be always stable. Several performance measures are computed. Some optimization problems are discussed. Gomez Corral, Krishnamoorthy and Viswanath (Gomez, Krishnamoorthy, and Narayanan 2005) analyzed a multi-server queueing system with a finite buffer and self-generation of priorities by waiting customers. Customers, at the time of joining the system, belong to one class and while waiting generate into priority. They provide formulae for numerical computation of variety of performance measures, including the blocking probability, the departure process and the stationary distributions of the system state at pre-arrival epochs, post arrival epochs and epochs at which arriving customers are lost.

In this paper, we consider a queueing system with $m+2$ priorities, labeled as $0,1,2,\ldots,m+1$, with 0 designated as super or highest priority and the rest of the $m+1$ priorities are such that type i has a higher priority over type j customers, for $i<j$. We assume that the customers arrive according to a Markovian arrival process with parametric representation given by (D_0,D_1) of order n. A brief description of MAP is given below. An arriving customer belongs to type i priority with probability p_i, $1\leq i\leq m+1$. The maximum capacity for priority type i is N_i with $N_0=1$, N_i is finite for $1\leq i\leq m$ and N_{m+1} is infinite. An arriving priority i customer has one of the following options. (a) Enters into service immediately due to the server being idle or busy with a customer whose priority type is lower than that of the arrival; (b) Enters into priority i buffer if the buffer is not full (unless $i=m+1$ in which case the customer will always enter into the buffer of infinite size); (c) Leaves the system since priority i buffer is full. Note that option (c) is not pertinent to type $m+1$ customers as there is no limit for how many such customers can be admitted. It is easy to see that the priority i customers arrive according to a Markovian arrival process with the arrivals governed by the matrix $\tilde{D}_i=p_iD_1$, $1\leq i\leq m+1$. In our model, we assume that the highest priority customers do not arrive externally. Instead these customers are generated randomly by the other types of priority customers while waiting for service. In general, the priority generation scheme is as follows. After a random amount of time that is exponentially distributed with parameter γ_{ii}, a priority type $i, 1\leq i\leq m+1$, customer waiting for service can turn into a highest priority with probability $\frac{\gamma_{i0}}{\gamma_{ii}}$ or into priority $j, j\neq i$, with probability $\frac{\gamma_{ij}}{\gamma_{ii}}$. Note that $\gamma_{ii}=\sum_{j\neq i,j=0}^{m+1}\gamma_{ij}$, for $1\leq i\leq m+1$, and that all priority $i, 1\leq i\leq m+1$, customers will independently change their priority type while waiting for service. We assume that while in service the customer cannot change the priority type. Furthermore, a highest priority customer finding another highest priority customer in service will leave the system without getting service and this is the reason why we take $N_0=1$. Thus, the set $\{\gamma_{ij}, 1\leq i\leq m+1, 0\leq j\leq m+1\}$, along with the numbers of customers of priority type $i, 1\leq i\leq m+1$ will completely describe the self-priority generating scheme.

The customers of priority type i have pre-emptive priority over customers of type j for $i<j$, $0\leq i,\ j\leq m+1$. The pre-empted customer will be served (as a new customer) when the service begins for this customer following the priority rule. The service time of a customer of priority type $i, 0\leq i\leq m+1$, is assumed to be of phase type (PH-type) with representation $(\alpha(i),S(i))$ and of order r_i, where $\alpha(i)$ is the initial probability vector. $S^0(i)$ is a column vector such that $S(i)e+S^0(i)=0$, where e is a column vector of 1's of appropriate order. Denoting by μ_i' to be the mean of $(\alpha(i),S(i))$, it can be verified that $\mu_i'=-\alpha(i)(S(i))^{-1}e$. Recall that a PH-distribution is the distribution of the time until absorption in a continuous time Markov chain with an absorbing state (Neuts 1981).

Observe that by taking $\gamma_{ij}=0$ for $1\leq i\leq m+1$; $0\leq j\leq m+1$, we get the classical priority queue. On the other hand letting $\gamma_{ij}\neq 0$ for $i\leq j$ and $\gamma_{ij}=0$ otherwise, the case of priority generation will be restricted to generating only one of higher priority. Similar interpretation (for generating one of lower priority) holds good when $\gamma_{ij}\neq 0$ for $j\geq i$, $i\neq 0$.

The *MAP* in continuous time is described as follows. Let the underlying Markov chain be irreducible and let $\hat{Q}$ be the generator of this Markov chain. At the end of a sojourn time in state i, that is exponentially distributed with parameter λ_i, one of the following two events could occur: with probability $r_{ij}(1)$ the transition corresponds to an arrival and the underlying Markov chain is in state j with $1\leq i,j\leq n$; with probability $r_{ij}(0)$ the transition corresponds to no arrival and the state of the Markov chain is j, $j\neq i$. Note that the Markov chain can go from state i to state i only through an arrival. Define matrices $D_0=(d_{ij}^0)$ and $D_1=(d_{ij}^1)$ such that $d_{ii}^0=-\lambda_i$, $1\leq i\leq m$, $d_{ij}^0=\lambda_i r_{ij}(0)$, for $j\neq i$ and $d_{ij}^1=\lambda_i r_{ij}(1)$, $1\leq i,j\leq n$. Note that for all $i, 1\leq i\leq n$, $\sum_{j=1}^{n}[r_{ij}(0)+r_{ij}(1)]=1$. By assuming D_0 to be a nonsingular matrix, the interarrival times will be finite with probability one and the arrival process does not terminate. Hence, we see that D_0 is a stable matrix. The generator $\hat{Q}$ is then given by $\hat{Q}=D_0+D_1$. If π is the steady-state probability vector of $\hat{Q}$, then the arrival rate λ is given by $\lambda=\pi D_1 e$, where e is a column vector of 1's of dimension n.

Thus, D_0 governs the transitions corresponding to no arrival and D_1 governs those corresponding to an arrival. It can be shown that *MAP* is equivalent to Neuts' versatile Markovian point process. The point process described by the *MAP* is a special class of semi-Markov processes. For further details on *MAP* and their usefulness in stochastic modeling, we refer to ((Lucantoni 1991), (Neuts 1989), (Neuts 1992)) and for a review and recent work on *MAP* we refer the reader to (Chakravarthy 2001).

The objective of this paper, therefore, is to generalize the classical priority queues. Of course we do this with restriction placed on the waiting spaces for all but the lowest priority one. This paper is presented as follows. In Section 2 we provide the description of the queueing model under study. In Section 3 we simulate this queueing model using ARENA, and discuss some illustrative numerical examples to qualitatively describe the model.

2 MODEL DESCRIPTION

The model described in the introduction can be studied using continuous time Markov chain as follows. First let $J_i(t)$, $0 \le i \le m+1$, denote the number of type i priority customers in the system at time t. Let $I_1(t)$ denote the state of the server with the convention that $I_1(t) = 0$ when the server is idle, and $I_1(t)$ will be the phase of the existing service when the server is busy at time t; and let $I_2(t)$ denotes the phase of the arrival process at time t. The process $\{H(t) : t \ge 0\}$ defined as $\{H(t) = (J_{m+1}(t), J_m(t), \ldots, J_1(t), J_0(t), I_1(t), I_2(t)) : t \ge 0\}$ is a continuous-time Markov chain with state space given by $\Omega = \cup_{k=0}^{\infty} l(k)$, where

$$l(0) = \{(0,0,\ldots,0,i_2)\,|1 \le i_2 \le n\} \cup \{(0, j_m, \ldots, j_0, i_1, i_2)| \quad 0 \le j_q \le N_q,\ 0 \le q \le m;\ \sum_{u=0}^{m} j_u \ne 0, 1 \le i_1 \le r_h, 1 \le i_2 \le n, h = min_{0 \le u \le m}\{j_u > 0\}\},$$

$$l(k) = \{(k, j_m, j_{m-1}, \ldots, j_1, j_0, i_1, i_2)\,| \quad 0 \le j_q \le N_q,\ 0 \le q \le m;\ 1 \le i_1 \le r_h, 1 \le i_2 \le n, h = min_{0 \le u \le m+1}\{j_u > 0\}\}, k \ge 1.$$

$l(k), k \ge 0$ is referred to as the level k containing set of states. Note that in $l(k)$ description it is obvious that $j_{m+1} = k$.

It is clear that the above continuous time Markov chain $\{H(t)\}$ is a level dependent quasi birth and death process (LDQBD). Arranging the states lexicographically the infinitesimal generator $Q = (q_{rs})$ of the process $\{H(t) : t \ge 0\}$ is given by

$$Q = \begin{bmatrix} A_{10} & A_{00} & & & & \\ A_{20} & A_{11} & A_0 & & & \\ & A_{22} & A_{12} & A_0 & & \\ & & A_{23} & A_{13} & A_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}.$$

The (block) matrices appearing in Q require extensive notations and since these are not absolutely necessary for our further discussion in this paper, the details are omitted here. Note that the system under discussion is always stable.

It may be noted that setting $\gamma_{m+1,j} = 0$ for $j = 0, 1, \ldots, m$, results in a level independent quasi birth and death process. In this case we get the infinitesimal generator of the process as a quasi-Toeplitz Markov chain and so the matrix geometric solution procedure can be adopted.

3 SIMULATION OF THE MODEL

The queueing model described in Section 2 is amenable to study through the classical algorithmic methods due to Neuts (Neuts 1981, Neuts 1989). However, we have chosen to simulate this queueing model using ARENA as the state space for the current model grows exponentially and the book-keeping becomes very intensive. Furthermore, the computation of the distributions of the waiting time in the system of various priority types (except the highest priority one) is very complicated to describe analytically. Thus, simulation will not only help to compare analytical results with those of the simulated ones, especially when one wants to generalize the model to include multiple-server case, but also get a feel for the waiting time distributions. The logic for developing the model is displayed in Figure 3 in the appendix. The ARENA modules for developing the model under study are displayed in Figures 4 and 5 in the appendix. The purpose of this section is to bring out the qualitative aspects of the queueing system under consideration through some interesting simulated (numerical) examples.
For our numerical discussions, we consider five different arrival processes with parameter matrices D_0 and D_1 given by

1. Erlang of order 3 (*ERL*)

$$D_0 = \begin{pmatrix} -3 & 3 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -3 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3 & 0 & 0 \end{pmatrix}$$

2. Exponential (*POI*):

$$D_0 = \begin{pmatrix} -1 \end{pmatrix}, D_1 = \begin{pmatrix} 1 \end{pmatrix}$$

3. Hyperexponential (HEX):

$$D_0 = \begin{pmatrix} -12.7 & 0 & 0 \\ 0 & -1.27 & 0 \\ 0 & 0 & -0.127 \end{pmatrix}, D_1 = \begin{pmatrix} 8.89 & 2.54 & 1.27 \\ 0.889 & 0.254 & 0.127 \\ 0.0889 & 0.0254 & 0.0127 \end{pmatrix}$$

4. *MAP* with negative correlation (*MNC*):

$$D_0 = \begin{pmatrix} -1.00222 & 1.00222 & 0 \\ 0 & -1.00222 & 0 \\ 0 & 0 & -225.75 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0.01002 & 0 & 0.9922 \\ 223.4925 & 0 & 2.2575 \end{pmatrix}$$

5. *MAP* with positive correlation (*MPC*):

$$D_0 = \begin{pmatrix} -1.00222 & 1.00222 & 0 \\ 0 & -1.00222 & 0 \\ 0 & 0 & -225.75 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0.9922 & 0 & 0.01002 \\ 2.2575 & 0 & 223.4925 \end{pmatrix}.$$

All these five *MAP* processes are normalized so as to have an arrival rate of 1. However, these are qualitatively different in that they have different variance and correlation structure. The first three arrival processes, namely *ERL*, EXP, and HEX, correspond to renewal processes and so the correlation is 0. The arrival process labelled *MNC* has correlated arrivals with correlation between two successive inter-arrival times given by -0.4889 and and the arrivals corresponding to the processes labelled *MPC* has a positive correlation with values 0.4889. The ratio of the standard deviations of the inter-arrival times of these five arrival processes with respect to *ERL* are, respectively, 1, 1.732051, 5.913554, 2.44136, and 2.44136.

For services we consider two cases:

- **ERL5:** Here we assume that all five priorities have the same service distribution given by Erlang of order 5. We take $\mu_i' = 0.9, 0 \leq i \leq 4$.
- **ERLV:** Here we assume that the five priorities have different service distributions with the highest priority having Erlang of order 5; customers of priority type i have Erlang of order $5-i, 1 \leq i \leq 4$. Here we normalize the rates in each phase of the five Erlangs such that $\mu_i' = 0.9, 0 \leq i \leq 4$.

Note that Erlang is a special case of a phase type distribution. It should be pointed out that Erlang is a built-in distribution in ARENA. However, we have provided the module in ARENA as displayed in Figure 2 for using phase type services (of order 3) as this is not available in ARENA.

In the following we will denote by γ_i the vector of dimension 5 that gives the rates of priority generation of a waiting priority $i, 1 \leq i \leq 4$ customer. That is, $\gamma_i = (\gamma_{i0}, \ldots, \gamma_{i4})$. In all our examples below we have fixed $\lambda = 1, \mu_i' = 0.9, 0 \leq i \leq 4, p_i = 0.25, 1 \leq i \leq 4$, and $\gamma_1 = (1.5, 5, 1.5, 1, 1), \gamma_2 = (2, 1.5, 6, 1.5, 2), \gamma_3 = (2, 1.5, 1, 5.5, 1), \gamma_4 = (3.5, 2, 2, 1.5, 9)$, $N_1 = 10, N_2 = 20$, and $N_3 = 30$.

We define the following measures (in steady state) for our discussion on the simulated results.

- $\mu_{WTS}^{(i)}, 0 \leq i \leq m+1$ the mean waiting time in the system of an admitted priority i customers (except for the priority $m+1$ customers who will always be admitted due to unlimited buffer size.
- $P_{busy}^{(i)}, 0 \leq i \leq m+1$ the probability that the server is busy with priority i customers at an arbitrary time.
- $P_{lost}^{(i)}, 0 \leq i \leq m$ the probability that a priority i customer will be lost due to lack of buffer space at an arbitrary time.
- $s_{WTS}^{(i)}, 0 \leq i \leq m+1$ the standard deviation of the waiting time in the system of an admitted priority i customer (except for the priority $m+1$ customers who will always be admitted due to unlimited buffer size.
- $\mu_{NQ}^{(i)}, 0 \leq i \leq m+1$ the mean number of priority i customers waiting in the queue at an arbitrary time.

For the above combinations of five arrival processes and two service schemes, we ran our simulation model for 10000 units and five replications. The above set of performance measures along with the half-widths of the intervals are displayed in Tables 1 and 2 below. Furthermore, three selected measures: log of the mean waiting time in the system of customers of various priorities, the probability that the server is busy with different types of customers, and the probability of a priority $i, 0 \leq i \leq 3$, customer is lost, are displayed in Figure 1.

A quick look at the entries in Tables 1, 2, and Figure 1, reveal the following observations.

1. The mean waiting time for the highest priority customer entering into service (remember that a highest priority customer may leave without getting service due to server being busy with another highest priority customer) is nothing but the mean service time and is assumed to be the same, namely 0.90, for all scenarios. The numbers are very close to 0.9. This fact can also be used as an accuracy check for simulated results.
2. The mean waiting time in the system for the lowest priority, namely, priority type 4, is largest among all customers and for all scenarios. This is as expected since these customers enter into the system without getting lost, and also are pre-empted by other priority customers.
3. It is interesting to see that in the case of renewal arrivals (namely, when comparing *ERL*,*POI*, and *HEX*), the probability that the server is busy with priority $i, 0 \leq i \leq 2$, customers appears to increase with increasing variability in the arrival process. However, the probability that the server is busy with priority $i, 3 \leq i \leq 4$, customers appears to decrease with increasing variability in the arrival process.
4. Looking at the probability that the server is busy with a specific priority type customer for the two correlated arrival processes, namely, *MNC* and *MPC*, we notice an interesting trend. For *MPC* process, this probability is larger as compared to *MNC* process in the case highest priority as well as for the lowest priority (priority 4) customers . However, for the other priority types, this probability is higher for *MNC* as compared to *MPC*.
5. We notice that a priority 3 customer has the highest probability of getting lost when the arrival process has a larger variability (*HEX*) or has a higher positively correlated arrivals (*MPC*). It is not surprising to see this phenomenon since these customers have a limited waiting space and are probably pre-empted more often (next only to priority 4 customers).
6. As is to be expected all the performance measures for renewal arrivals, namely, for *ERL*, *POI*, and *HEX* arrival processes, behave similar to what is to be expected. For example, the mean and standard deviation of the waiting times tend to increase with increasing variability.
7. While *MNC* and *MPC* arrival processes have the same mean and standard deviation, yet some of the key measures such as the mean waiting times in the system of priority $i, 1 \leq i \leq 4$, customers are significantly different. This indicates the crucial role played by the correlated, especially the positive one, arrivals in stochastic modeling. We have seen such a crucial role played by the correlated arrivals in our other stochastic models analyzed using analytical and computational modeling tools.

Now we look at the fitted distributions of the waiting time distributions of various types of customers. Using the simulated data, ARENA has the option to identify the best fit (based on the least sum of squares due to error) among many distributions. In Tables 3 and 4 we list the fitted distributions for the various scenarios considered. Some sample histograms of simulated data along with the fitted distributions are displayed in Figure 2. An examination of these tables reveal the following.

- As expected an admitted highest priority customer's waiting time in the system is nothing but the service time distribution which is assumed to be Erlang of order 5.
- It is worth noting that the waiting time distribution of a priority 4 customer is fitted using beta distribution for all scenarios. As is known, the beta distribution has the ability to fit a variety of shapes in the data, and due to priority 4 customers having more chances of getting pre-empted by other types of customers and they all leave the system only after getting a service, the waiting times have more variability.
- Normally the waiting time distribution in a queueing model is skewed to the right (to accommodate for some customers having to wait unusually longer than the others) and this can be seen in ARENA identifying lognormal distribution to be the best fit for many combinations.

In conclusion, we have shown how simulation (using powerful software such as ARENA) can be used to bring out the qualitative aspect of a complicated stochastic model. Further research is currently being done to incorporate interesting optimization problems and the results of which will be presented elsewhere.

Table 1: Selected performance measures for various arrival processes with identical services

PERFORMANCE MEASURES	*ERL*		*POI*		*HEX*		*MNC*		*MPC*	
	Average	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$
$\mu_{WTS}^{(0)}$	0.90756	0.02792	0.89548	0.01722	0.90422	0.00946	0.89749	0.00987	0.89685	0.01347
$\mu_{WTS}^{(1)}$	1.08160	0.01754	1.14330	0.01452	2.14750	0.13259	1.21680	0.02993	2.04990	0.10389
$\mu_{WTS}^{(2)}$	1.73220	0.04678	2.37880	0.08655	15.19600	2.01650	2.80730	0.22619	9.86290	1.01300
$\mu_{WTS}^{(3)}$	8.77580	0.30126	18.20300	4.91690	100.29000	14.77000	31.60900	9.08760	38.52200	7.68620
$\mu_{WTS}^{(4)}$	2877.3	52.1	3536.1	460.5	4434.8	581.5	3653.8	395.7	1636.6	1173.7
$P_{busy}^{(0)}$	0.08855	0.00206	0.10793	0.00341	0.16838	0.00439	0.11936	0.00414	0.14368	0.00980
$P_{busy}^{(1)}$	0.28595	0.00316	0.29775	0.00566	0.33614	0.01369	0.31028	0.01039	0.23606	0.00612
$P_{busy}^{(2)}$	0.27233	0.00454	0.27188	0.00791	0.28493	0.00993	0.27736	0.00678	0.22723	0.00476
$P_{busy}^{(3)}$	0.25160	0.00318	0.24732	0.00523	0.18224	0.01509	0.24858	0.00606	0.19546	0.00856
$P_{busy}^{(4)}$	0.10157	0.00215	0.07512	0.01148	0.02831	0.01466	0.04442	0.01725	0.19758	0.01022
$P_{lost}^{(0)}$	0.00002	0.00005	0.00002	0.00005	0.00007	0.00009	0.00004	0.00006	0.00002	0.00004
$P_{lost}^{(1)}$	0.00000	0.00000	0.00000	0.00000	0.00015	0.00018	0.00000	0.00000	0.05685	0.01862
$P_{lost}^{(2)}$	0.00000	0.00000	0.00000	0.00000	0.01263	0.00461	0.00000	0.00000	0.05121	0.02293
$P_{lost}^{(3)}$	0.00000	0.00000	0.00022	0.00037	0.11258	0.02405	0.00191	0.00177	0.06803	0.02245
$P_{lost}^{(4)}$	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
$s_{WTS}^{(0)}$	0.40518	0.00539	0.40008	0.00953	0.40220	0.00943	0.39840	0.01183	0.39609	0.01138
$s_{WTS}^{(1)}$	0.60279	0.04021	0.67808	0.02943	2.06340	0.21219	0.77300	0.04281	3.68170	1.19800
$s_{WTS}^{(2)}$	1.49160	0.12466	2.36370	0.21876	18.60100	10.53000	2.66550	0.22961	14.65800	0.58972
$s_{WTS}^{(3)}$	10.52700	0.55645	18.41000	4.79220	80.58800	12.86400	26.69500	6.10350	49.29200	6.77910
$s_{WTS}^{(4)}$	1703.900	61.895	1943.400	126.670	2343.400	402.370	2511.500	424.850	706.060	446.110
$\mu_{NQ}^{(0)}$	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
$\mu_{NQ}^{(1)}$	0.05449	0.00370	0.07567	0.00391	0.43384	0.06275	0.10087	0.00848	0.27306	0.05099
$\mu_{NQ}^{(2)}$	0.16401	0.01324	0.33125	0.03108	4.17350	0.74292	0.45089	0.05529	2.10270	0.39690
$\mu_{NQ}^{(3)}$	1.81620	0.09616	4.29360	1.35590	19.60900	1.76530	7.92000	2.37130	7.85640	1.54490
$\mu_{NQ}^{(4)}$	810.90	27.29	1044.10	70.78	1605.20	111.44	1256.90	106.43	442.16	340.15
Utilization	0.99930	0.00060	0.99918	0.00100	0.99828	0.00236	0.99977	0.00024	0.97646	0.05842

Table 2: Selected performance measures for various arrival processes with different service distributions

PERFORMANCE MEASURES	*ERL*		*POI*		*HEX*		*MNC*		*MPC*	
	Average	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$	*Average*	$\frac{1}{2}width$
$\mu_{WTS}^{(0)}$	0.90426	0.01429	0.89510	0.00786	0.90251	0.01187	0.88986	0.02266	0.90120	0.01712
$\mu_{WTS}^{(1)}$	1.07530	0.01838	1.17430	0.02514	2.16110	0.08350	1.20870	0.02397	2.22400	0.19361
$\mu_{WTS}^{(2)}$	1.84300	0.07403	2.55730	0.13963	14.35600	1.52340	2.84320	0.10748	11.72200	1.51640
$\mu_{WTS}^{(3)}$	10.09300	2.15960	27.49300	6.41370	94.50000	11.50100	27.45300	3.48120	45.58200	6.89320
$\mu_{WTS}^{(4)}$	2903.70000	468.33000	3760.80000	367.97000	3946.40000	831.87000	4152.80000	806.50000	2099.40000	650.68000
$P_{busy}^{(0)}$	0.09392	0.00472	0.11471	0.00500	0.16913	0.00307	0.11973	0.00434	0.14689	0.00383
$P_{busy}^{(1)}$	0.28498	0.00415	0.30032	0.00635	0.33009	0.01063	0.30500	0.00860	0.24706	0.01132
$P_{busy}^{(2)}$	0.27076	0.00237	0.27672	0.00642	0.28172	0.00814	0.27497	0.00402	0.23265	0.00583
$P_{busy}^{(3)}$	0.24886	0.00521	0.24935	0.00497	0.18578	0.01407	0.24990	0.00583	0.20321	0.00324
$P_{busy}^{(4)}$	0.10148	0.01073	0.05890	0.01270	0.03327	0.00703	0.05041	0.00563	0.17019	0.02161
$P_{lost}^{(0)}$	0.00020	0.00020	0.00014	0.00017	0.00022	0.00006	0.00022	0.00013	0.00005	0.00005
$P_{lost}^{(1)}$	0.00000	0.00000	0.00000	0.00000	0.00041	0.00038	0.00000	0.00000	0.06572	0.01012
$P_{lost}^{(2)}$	0.00000	0.00000	0.00000	0.00000	0.01027	0.00201	0.00000	0.00000	0.06205	0.01065
$P_{lost}^{(3)}$	0.00002	0.00005	0.00215	0.00235	0.10328	0.01841	0.00170	0.00163	0.07942	0.01135
$P_{lost}^{(4)}$	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
$s_{WTS}^{(0)}$	0.39870	0.01141	0.39606	0.01114	0.40492	0.01747	0.39879	0.00654	0.39831	0.01327
$s_{WTS}^{(1)}$	0.66169	0.10778	0.77892	0.04781	2.46970	0.94459	0.78382	0.03022	3.88800	2.62360
$s_{WTS}^{(2)}$	1.75820	0.13052	2.67620	0.30742	13.49300	0.83433	3.29880	1.22750	15.84900	1.04690
$s_{WTS}^{(3)}$	12.37900	3.79110	26.75200	6.29030	78.77300	14.04900	25.01700	3.62690	50.46400	3.23130
$s_{WTS}^{(4)}$	1693.30000	282.89000	2180.40000	182.61000	2496.50000	342.72000	2381.50000	169.06000	1093.50000	454.74000
$\mu_{NQ}^{(0)}$	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
$\mu_{NQ}^{(1)}$	0.05255	0.00395	0.08497	0.00544	0.42781	0.04049	0.09868	0.00704	0.34065	0.05452
$\mu_{NQ}^{(2)}$	0.19624	0.01983	0.38750	0.04103	3.92510	0.56637	0.45598	0.03062	2.66370	0.42103
$\mu_{NQ}^{(3)}$	2.15810	0.60056	6.85050	1.78490	18.66400	1.13870	6.83620	0.89532	9.83460	1.56240
$\mu_{NQ}^{(4)}$	824.81000	109.18000	1160.30000	63.96900	1539.20000	98.22300	1246.60000	78.94900	632.31000	230.81000
Utilization	0.99916	0.00207	0.99960	0.00108	0.99696	0.00355	0.99965	0.00043	0.99380	0.00806

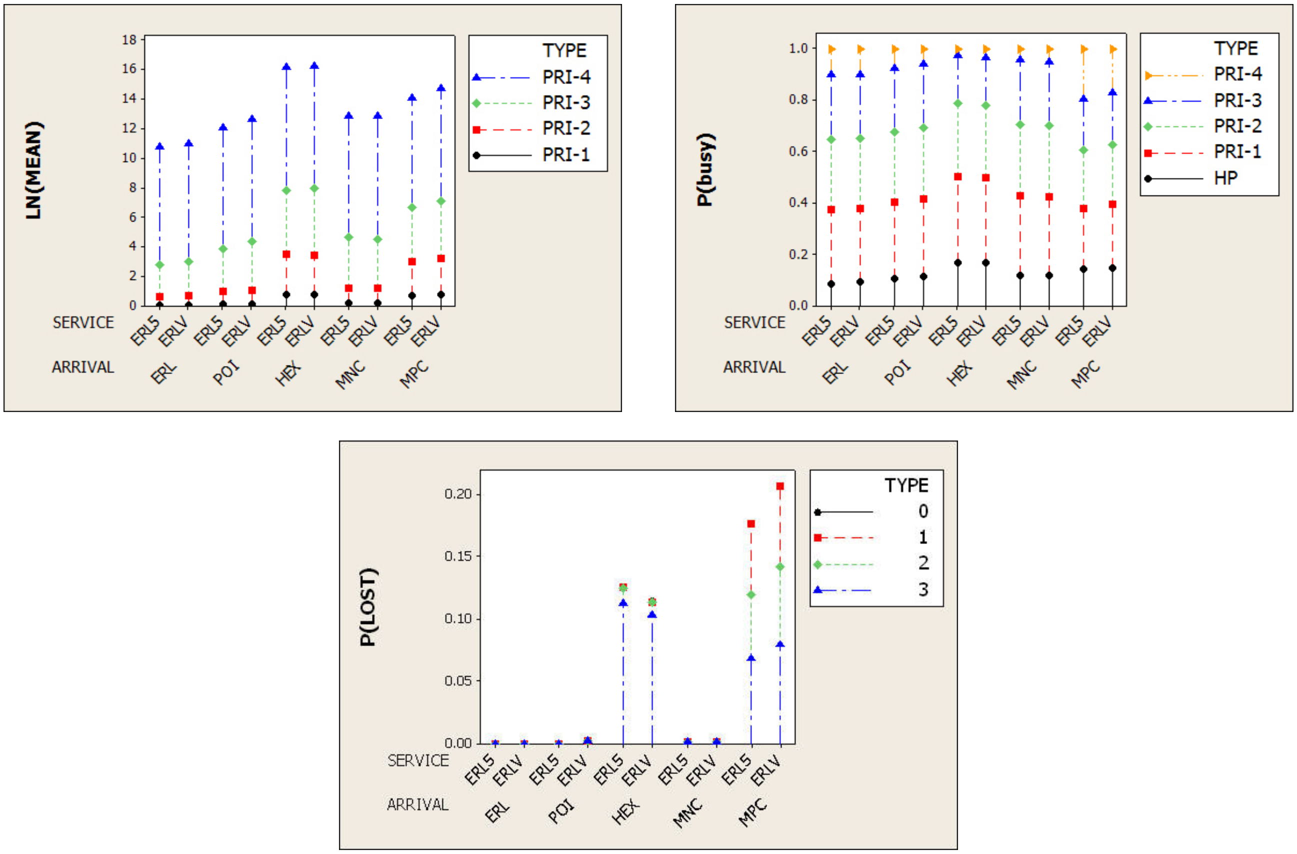

Figure 1: Comparison of selected measures for various scenarios

Table 3: Identification of the fitted distributions for the waiting time in the system for identical services

ARRIVAL	Highest Priority	Priority 1	Priority 2	Priority 3	Priority 4
ERL	$ERLANG(0.181,5)$	$LOGN(1.08,0.607)$	$LOGN(1.71,1.41)$	70 $BETA(0.481,3.36)$	6060 $BETA(0.933,1.06)$
POI	$ERLANG(0.179,5)$	$LOGN(1.14,0.676)$	$LOGN(2.34,2.34)$	114 $BETA(0.607,3.19)$	7270 $BETA(1.15,1.21)$
HEX	$ERLANG(0.181,5)$	$LOGN(2.12,2.11)$	$EXP(15.2)$	492 $BETA(0.724,3.83)$	9310 $BETA(1.61,2.12)$
MNC	$ERLANG(0.18,5)$	$LOGN(1.22,0.766)$	$LOGN(2.81,2.93)$	178 $BETA(0.865,4.03)$	8680 $BETA(0.805,1.14)$
MPC	$ERLANG(0.179,5)$	$WEIBULL(2.07,0.935)$	$LOGN(10.2,35.1)$	$LOGN(73.4,697)$	3990 $BETA(0.802,1.12)$

Table 4: Identification of the fitted distributions for the waiting time in the system for varying services

ARRIVAL	Highest Priority	Priority 1	Priority 2	Priority 3	Priority 4
ERL	$ERLANG(0.181,5)$	$GAMMA(0.324,3.32)$	$LOGN(1.85,1.87)$	98 $BETA(0.458,3.98)$	6380 $BETA(1.07,1.3)$
POI	$ERLANG(0.179,5)$	$LOGN(1.18,0.79)$	$LOGN(2.58,3.09)$	146 $BETA(0.622,2.68)$	8300 $BETA(1.17,1.42)$
HEX	$ERLANG(0.181,5)$	$EXPO(2.16)$	291 $BETA(1.02,19.6)$	926 $BETA(1.164,10.3)$	8670 $BETA(0.83,0.978)$
MNC	$ERLANG(0.178,5)$	$LOGN(1.21,0.818)$	$GAMMA(0.823,4.22)$	168 $BETA(0.823,4.22)$	8510 $BETA(0.991,1.09)$
MPC	$ERLANG(0.18,5)$	$GAMMA(2.17,1.03)$	$LOGN(11.9,43.6)$	$LOGN(98.6,824)$	5490 $BETA(1.6,2.76)$

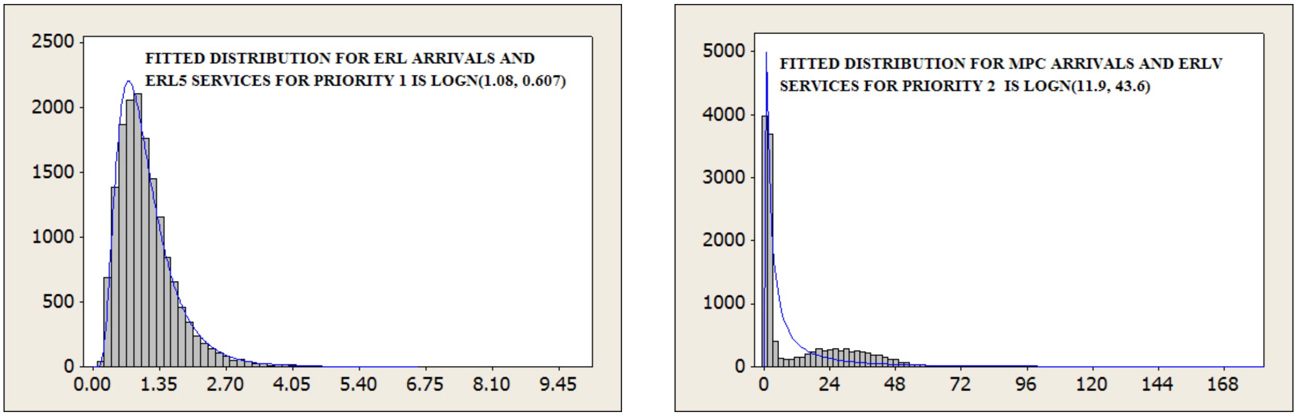

Figure 2: Histograms and fitted distributions for waiting time in the system for selected scenarios

REFERENCES

Chakravarthy, S. R. 2001. The batch Markovian arrival process: A review and future work. In *Advances in Probability Theory and Stochastic Processes.*, ed. A. K. et al., 21–39. New Jersey: Notable Publications Inc.,.

Gomez, C. A., A. Krishnamoorthy, and V. C. Narayanan. 2005. The impact of self-generation of priorities on multi-server queues with finite capacity. *Stochastic Models* 21 (2–3): 427–447.

Jaiswal, N. 1968. *Priority queues*. New York: Academic Press.

Krishnamoorthy, A., T. Deepak, and V. C. Narayanan. 2002. Queues with self-generation of priorities. Technical report, Cochin University of Science and Technology.

Krishnamoorthy, A., V. C. Narayanan, and T. G. Deepak. 2005. On a queueing system with self generation of priorities. *Journal of Neural Parallel and Scientific Computations*.

Lucantoni, D. 1991. New results on the single server queue with a batch markovian arrival process. *Stochastic Models* 7:1–46.

Neuts, M. 1989. *Structured stochastic matrices of m/g/1 type and their applications*. NY: Marcel Dekker.

Neuts, M. 1992. Models based on the markovian arrival process. *IEICE Transactions on Communications* E75B:1255–1265.

Neuts, M. F. 1981. *Matrix-geometric solutions in stochastic models: An algorithmic approach*. Baltimore, MD: The Johns Hopkins University Press (Dover since 1994).

Takagi, H. 1989. *Queueing analysis - volume 1: Vacations and priority systems*. Amsterdam: North-Holland.

Wang, Q. 2004. Modeling and analysis of high risk patient queues. *European J. Oper. Res.* 155:502–515.

AUTHOR BIOGRAPHIES

A. KRISHNAMOORTHY is Professor of Applied Mathematics at the Cochin University of Science & Technology, India, since January 1987. He received the Doctoral degree in 1978 and has over 100 research publications and has edited a few conference proceedings. He is Editor of OPSEARCH and is on the Editorial board of a number of journals. His areas of interest include Operations Research, Stochastic Modelling with special reference to Queues, Inventory and Reliability. His email address for these proceedings is <ak@cusat.ac.in>.

VISWANATH C. NARAYANAN is Lecturer in mathematics at Government Engineering College, Trissur, India. He has over 20 research publications. His areas of interest include Stochastic modeling in Queues, Inventory, Reliability and Bio-informatics. His email address for these proceedings is <viswam@cusat.ac.in>.

SRINIVAS R. CHAKRAVARTHY is a Professor of Operations Research and Statistics in the Department of Industrial and Manufacturing Engineering at Kettering University, Flint, Michigan, USA. His research interest are in the areas of algorithmic probability, queuing, reliability and inventory. He has published more than 70 papers in leading journals and presented papers at national and international conferences. He co-organized the First and Second International Conferences on Matrix-analytic Methods in Stochastic Models in 1995 and 1998. He received Kettering University Alumni's Outstanding Teacher Awards (1990 and 2001) and Kettering University's Outstanding Researcher (1996) and Distinguished Researcher (2003) Awards. He is on the scientific advisory committee of several international conferences, and also on the editorial board of a few journals. His email address for these proceedings is <schakrav@kettering.edu>.

APPENDIX

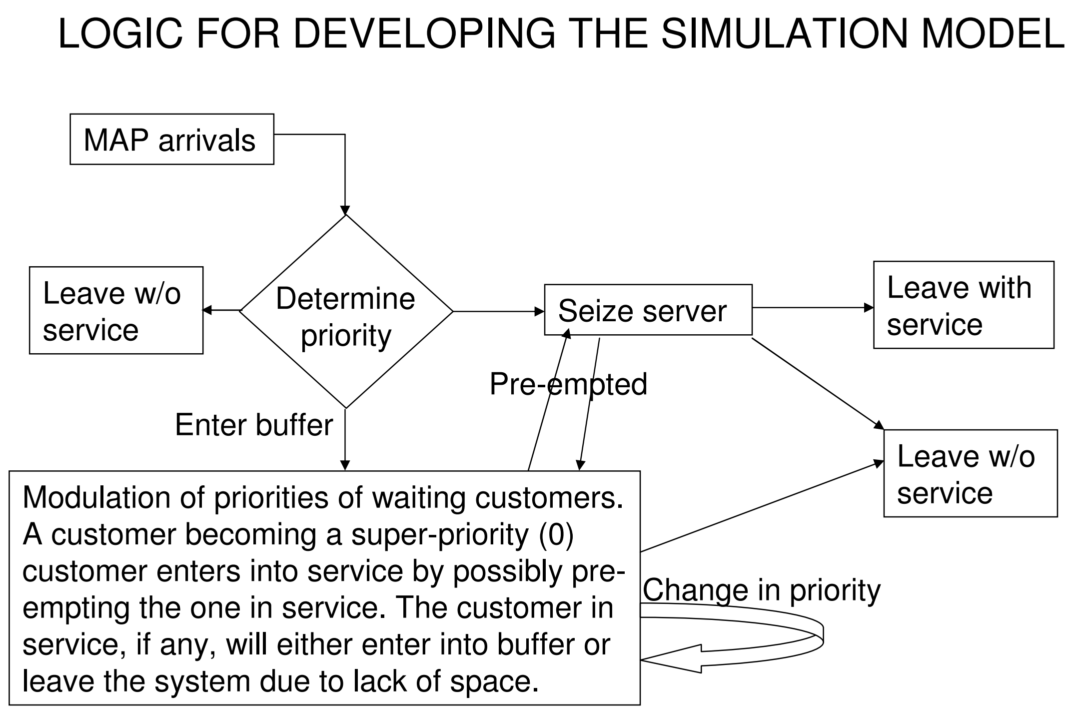

Figure 3: Logic for the development of ARENA model

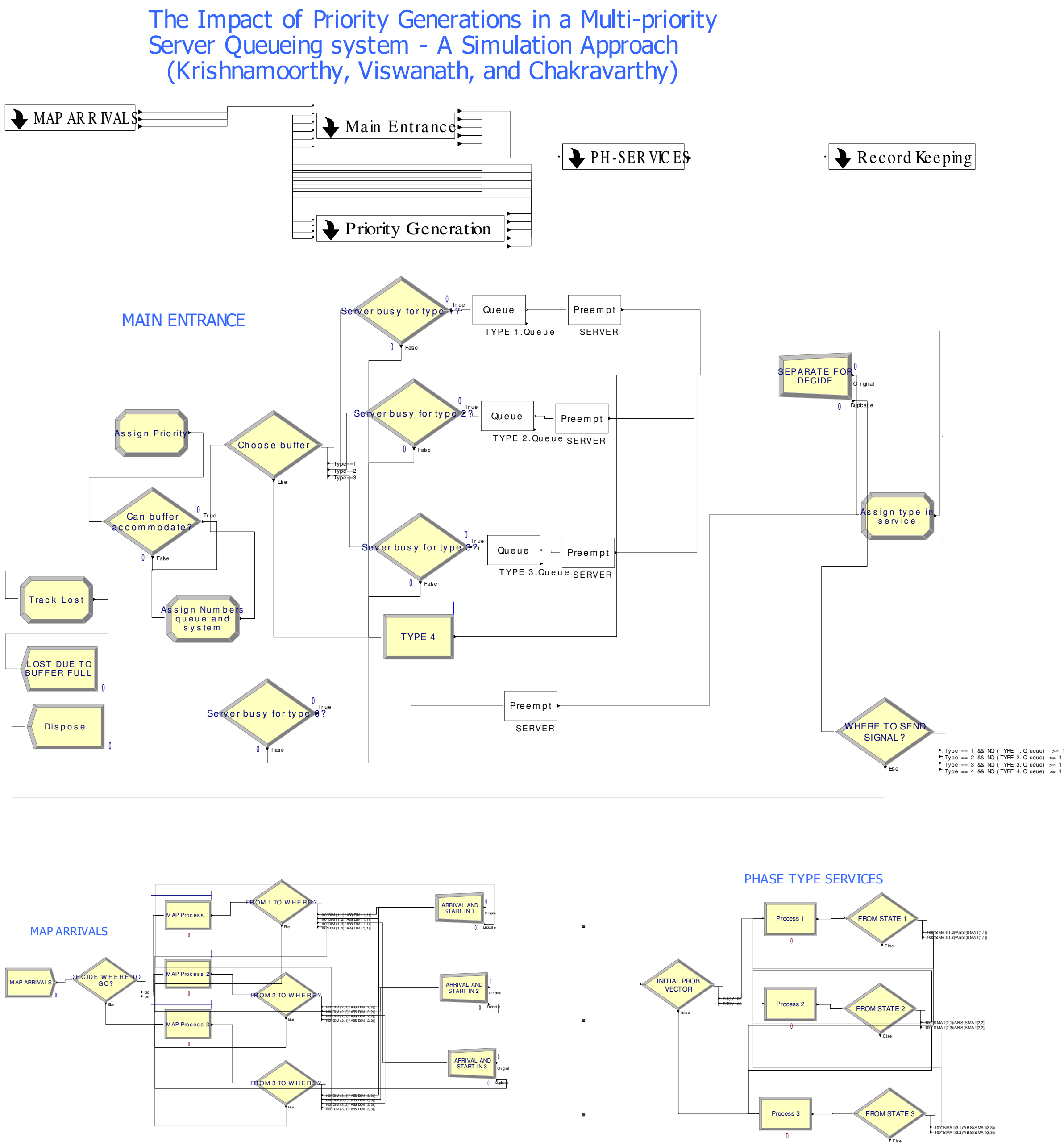

Figure 4: ARENA MODULES - Main, MAP arrivals and Phase type services

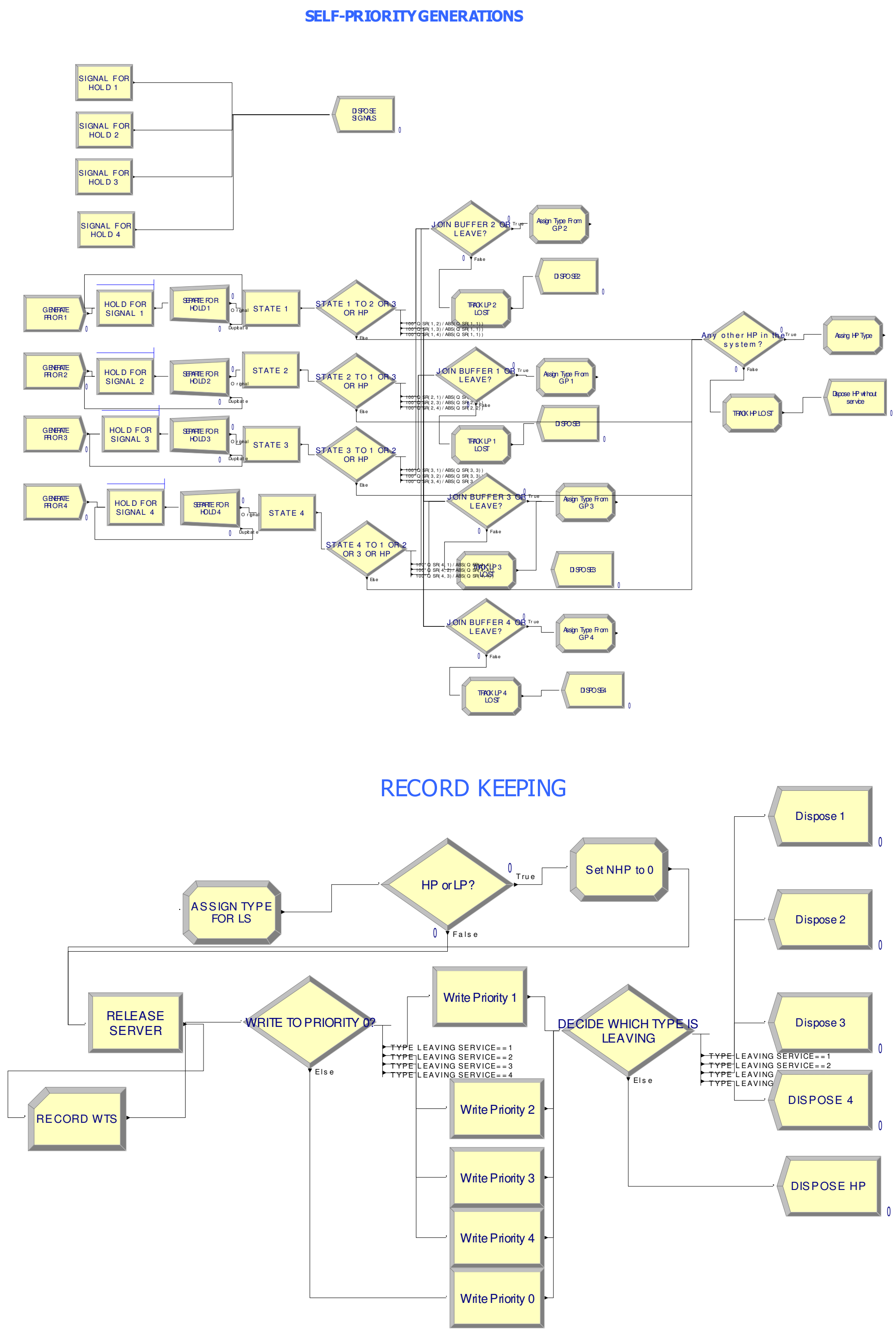

Figure 5: ARENA MODULES - Self-priority generation and Record keeping