ISSN 0005-1179, Automation and Remote Control, 2012, Vol. 73, No. 2, pp. 404–407. © Pleiades Publishing, Ltd., 2012.
Original Russian Text © S.V. Titov, A.V. Burkovskii, 2008, published in Sistemy Upravleniya i Informatsionnye Tekhnologii, 2008, No. 2, pp. 407–410.

Agent-based Simulation Modeling Technology for Queueing Networks

S. V. Titov and A. V. Burkovskii

Voronezh State Technical University, Voronezh, Russia

Received July 7, 2008

Abstract—In this paper we consider a modeling approach to queueing systems implementing an essentially network structure; requests in the network have an individual mode of behavior.

DOI: 10.1134/S0005117912020191

Let us study a modeling problem for queueing systems with service requests having individual modes of behavior. An example could be provided by polyclinics, transport networks, multiphase manufacturing systems, etc [1]. In particular, the described class definitely includes systems, where service requests are actually represented by humans or automatic (semiautomatic) intelligent objects. A specific feature of such systems consists in that the requests possess a certain behavior; thus, one would say that "intelligent" requests take place. For instance, in a polyclinic patients belonging to different nosological groups (hypertensive or cardiovascular patients) visit numerous doctors and medical treatment rooms. Note it seems impossible to specify their routes rigorously; the number of unique routes of the patients depends on the actual number of the requests [2]. On the other hand, an "intelligent" nature of the requests prevents from defining the routes of the request flows as transition-probability matrices. Indeed, *in lieu of* passing from the ith device to the jth one (with probability p_{ij}), a request may "prefer" to visit another device depending on some additional conditions (e.g., due to a long queue before the jth device).

In the sequel we develop a modeling technology for such systems; as an example, let us consider the following problem of evaluating performance characteristics of an open-loop queueing network operating in a stationary mode. The network is composed of R nodes, and several requests of Q types circulate among them. Each type has a corresponding priority d_j $(1 \leq j \leq Q)$, while the ith node $(1 \leq i \leq R)$ contains m_i parallel service channels. There is a queue with l_i waiting positions for the corresponding servicing device. We emphasize that FIFO principle serves for queueing, and the procedure of absolute priorities is employed (a request with higher priority is placed before its counterpart with smaller priority). For all types of the requests and any device of the ith node, servicing time may have an arbitrary distribution. Interaction scheme of individual nodes within the network is determined by a fully connected graph, where the edge s_{ij} links the R_ith node to the R_jth one. Finally, Q flows of the requests enter the network; intervals between the adjacent requests possess an arbitrary distribution law.

Generally, a route of requests in a queueing system with several classes is defined by a matrix

$$P = \|p_{ir,k}\| \,. \tag{1}$$

Here $P_{ir,k}$ stands for probability of the event that the rth-class request (after being serviced at the ith node) passes to the kth one.

However, this is inadmissible due to individual character of the requests. For each type of the requests, their route depends on necessity to visit a specific set of servicing devices (with the given probability). Moreover, composition of the set and the underlying probabilities of visiting the servicing devices are subject to the type of the request. Therefore, to describe interaction between nodes of the network, we introduce the following matrix:

$$P' = \begin{pmatrix} p'_{1,1} & p'_{2,1} & \cdots & p'_{m,1} \\ p'_{1,2} & p'_{2,2} & \cdots & p'_{m,2} \\ \cdots & \cdots & \ddots & \cdots \\ p'_{1,n} & p'_{2,n} & \cdots & p'_{m,n} \end{pmatrix}. \tag{2}$$

In the previous formula, $p'_{i,j}$ means probability of the event that a request of the ith type visits the jth servicing device.

Classical discrete-event modeling of queueing networks (QN) with several types of requests operates a multidimensional state vector of the network. In other words, at each instant the state of the ith node is given by a vector $N_i(t) = (N_{i1}(t), N_{i2}(t), \ldots, N_{in_i}(t)))$; $N_{ij}(t)$ denotes the number of the request type which holds the jth position in the ith node, $1 \le N_{ij}(t) \le Q$, $1 \le i \le R$, $1 \le j \le n_i$, $n_i \ge 1$. The ith node being free at an instant t, one sets $N_i(t) = 0$. However, such approach is inapplicable to the model considered due to individual character of the requests.

To solve the above-mentioned difficulties, we will adopt an agent-request; it possesses a certain memory, includes a list of servicing devices that have been visited by the request and contains additional information (arrival time of the request, leaving time from the device, etc). An important advantage of the agent-based framework lies in feasibility of accumulating a detailed statistics. The matter is that after the modeling process every request keeps in the memory its route through the nodes.

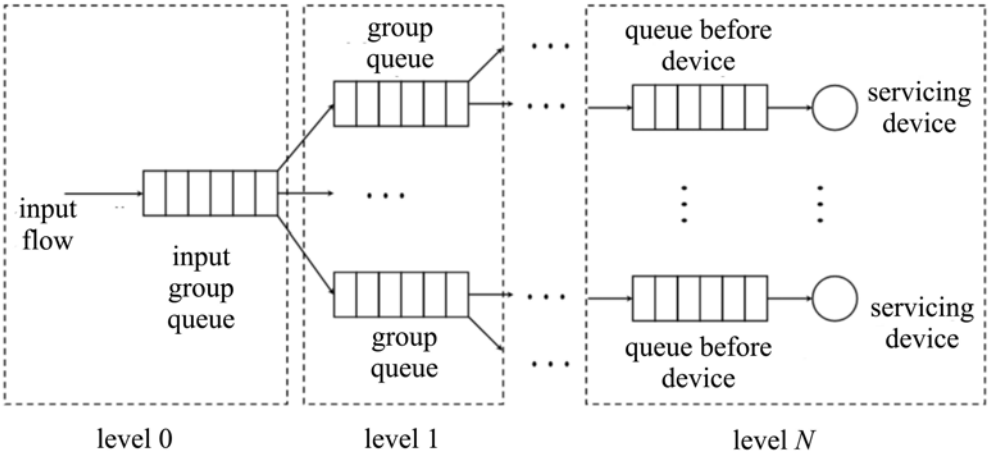
During generation of the next request, the matrix P' makes it possible to define a visit vector of the servicing devices, $\bar{B} = \{b_1, b_2, \ldots, b_R\}$. Here the component b_i makes 0 if the current request must not visit the ith servicing device in the future (and it constitutes 1 otherwise). As soon as the request has been processed by the ith servicing device, the corresponding value b_i must be changed from 1 to 0. Consequently, all elements of the vector $\bar{B}$ being zero forms the service completion condition for the request considered. And the latter is added to the flow of processed requests.

Let us also consider the following agents: an input flow, output flows of processed and unprocessed requests, a queue, a multichannel servicing device, a relation between objects. In the sequel we study their interaction and functions.

Interaction of the nodes within QN is described as follows. An input flow of requests relates to an input group queue. This queue is possibly connected to other group queues of the subsequent nesting level; thus, a multilevel tree-type structure appears. It ends with queues before certain servicing devices. A graphical scheme of such structure is illustrated by figure.

The "input flow" agent is responsible for generating Q different flows of requests which correspond to Q request types of the model (this is the case on the strength of independent arrival time distributions used for different types of requests). Furthermore, the input flow performs dispatching of the requests to the input group queue or to the output flow of unprocessed requests (if all waiting positions in the input group queue are occupied). The output flows of processed and unprocessed requests represent a storage of the agent-requests that have been removed from the system (as the result of rejection or successfully completed servicing). These sets would be utilized in a statistical analysis of the modeling results.

The servicing devices are combined in groups subject to certain criteria existing in the physical system modeled (e.g., in terms of functions or location). Hence, it seems reasonable for a request to pass all devices of the group (it belongs to), and then be processed by other servicing devices. Within the structure of the model, all groups of the devices are formally implemented via group queues; i.e., a request tries to visit all (necessary) servicing devices that have been added to the



A tree-type structure of the model.

queue considered. Using the matrix $\overline{B}$, we obtain transition probabilities of the request (from a certain state to another one) as follows. Denote by $\overline{E} = \{e_1, e_2, \ldots, e_R\}$ a membership vector of servicing devices to the queue which includes the request (at the moment in question). Then e_i equals 1 if the device belongs to the queue (and 0 otherwise). Therefore, probability of request's visiting the ith device is defined by

$$p_i'' = \begin{cases} \frac{b_i e_i}{\sum\limits_{j=1}^{n} b_j e_j}, & \text{if} \quad \sum\limits_{i=1}^{n} b_i e_i \neq 0 \\ \frac{b_i}{\sum\limits_{j=1}^{n} b_j}, & \text{if} \quad \sum\limits_{i=1}^{n} b_i e_i = 0. \end{cases} \tag{3}$$

The relation is an auxiliary object reflecting moving direction of the flows of agent-requests. There are two kinds of relations, *viz.*, positive and negative ones. The former indicates a receiving agent which gets the requests from a sending agent (in the case of successful servicing); the latter determines a receiving agent which gets the rejected request. Positive relation being missed at the output of a certain object, the object in question is assumed to be related to the output flow of processed requests. Similarly, all negative relations are (on default) directed to the output flow of unprocessed requests.

A servicing device includes parameters of the servicing time distribution, as well as an array of channels. Each element of the array represents a sample of the servicing channel and contains total hold-off time of the channel; at any instant the channel is busy, the element also stores the agent-request, arrival time of the request, and estimated time of service completion. The behavior of the servicing device consists in accepting a new request for processing and putting it to a free (available) channel. The behavior also includes dispatching of the leaving (successfully serviced) request to a destination agent (depending on the output relations of the device), as well as updating the states of the channels at each modeling stage.

The channels of servicing devices are supposed parallel. During modeling, the first channel is occupied initially, then the second one and so on. In other words, the program involves priorities of the channels. However, in real life the channels possess identical priorities; thus, a random free channel is occupied. In this paper we proceed from the concept of different-priority channels due to the following reasons. First, imagine a servicing device represents a black box with respect to the remaining part of the model; then the behavior of the servicing device is the same regardless of priorities of the channels and depends on their number. Second, under identical priorities we would not be able to answer several questions: "To what extent do we need the ith channel?," "Is it reasonable to add one or more channels?" Instead, we merely know the answer to the question, "What is the average load of the channels?" On the other hand, the first couple of questions are easily clarified for the channels with different priorities. Moreover, the average load of the equivalent channels (in the sense of priorities) is given by a simple formula

$$K'' = \frac{\sum\limits_{i=1}^{N} K_i'}{N}. \tag{4}$$

Here

K'' means the load level of the channels with identical priorities;

K_i' stands for the load level of the ith channel with a different priority;

N is the number of the channels.

Finally, we underline that each object engaged in dispatching the output requests (e.g., input flows, servicing devices) is related to the output flows of processed and unprocessed requests. Assume a certain request could not be moved to a destination object (for instance, the proper servicing device is busy or all waiting positions are occupied). Then the request will be directed to the output flow of unprocessed requests (and accounted as a rejected one in the corresponding statistics). If the request completes processing (notably, all elements of the vector $\bar{B}$ make zero), it passes to the flow of processed requests for a further statistical treatment.

The discussed approach allows for better adequacy in reflecting the real processes in queueing networks.

REFERENCES

1. Burkovskii, V.L., Titov, S.V., and Burkovskii, A.V., *Imitatsionnoe modelirovanie i optimizatsiya setei massovogo obsluzhivaniya na osnove evolyutsionnykh metodov* (Simulation Modeling and Optimization of Queueing Networks Based on the Evolution Methods), Voronezh: Gos. Tekhn. Univ., 2007.

2. Burkovskii, V.L. and Titov, S.V., Simulation Modeling of a Polyclinic Based on Open-loop Stochastic Networks, *Vestn. Voronezh. Gos. Tekhn. Univ.*, 2002, vol. 8.2, pp. 86–88.