# Tool supported modeling of sensor communication networks by using finite-source priority retrial queues *

Tamás Bérczes, János Sztrik, Péter Orosz
Faculty of Informatics, University of Debrecen
Debrecen, Hungary,
{berczes.tamas,sztrik.janos,orosz.peter}@inf.unideb.hu
Pascal Moyal, Nicolaos Limnios, Stylianos Georgiadis
Universite de Technologie de Compiegne
Compiegne, France,
{pascal.moyal,nikolaos.limnios}@utc.fr
stylianos.georgiadis@hds.utc.fr

## Abstract

**The main aim of the present paper is to draw the attention of the readers of this special issue to the modeling issues of sensor networks. The novelty of this investigation is the introduction of servers vacation combined with priority customers for finite-source retrial queues and its application to wireless sensor networks. In this paper we analyze a priority finite-source retrial queue with repeated vacations. Two types of priority customers are defined, customers with priority 1 ($P_1$) go directly to an ordinary FIFO queue. However, if customers with priority 2 ($P_2$) find the server in busy or unavailable state go to the orbit. These customers stay in the orbit and retry their request until find the server in idle and available state. We assume that $P_1$ customers have non-preemptive priority over $P_2$ customers. The server starts with a listening period and if no customer arrive during this period it will enter in the vacation mode. When the vacation period is terminated, then the node wakes up. If there is a $P_1$ customer in the queue the server begin to serve it, and when there is no any $P_1$ customer, the node will remain awake for exponentially distributed time period. If that period expires without arrivals the node will enter in the next sleeping period. All random variables involved in model construction are supposed to be independent and exponentially distributed ones. Our main interest is to give the main steady-state performance measures of the system computed by the help of the MOSEL tool. Several Figures illustrate the effect of input parameters on the mean response time.**

**Keywords.** Performance modeling; Finite-source retrial queues; Sensor networks; MOSEL

## I. INTRODUCTION

Retrial queues have been widely investigated and used to model many problems arising in telephone switching systems, telecommunication networks, computer networks, optical networks and most recently sensor networks, etc. The main characteristic of a retrial queue is that a customer who finds the service facility busy upon arrival is obliged to leave the service area, but some time later he comes back to re-initiate his demand. Between trials a customer is said to be in orbit. The literature on retrial queueing systems is very extensive. For a recent account, readers may refer to the recent books of Artalejo and Gomez-Corral [2] and Falin and Templeton [13] that summarize the main models and methods. For some recent results on retrial queues with applications the interested reader is referred to, for example [3], [8],[9] and references therein.

Priority retrial queues have been studied by many researchers so far. High priority customers are queued and served according to some discipline. In case of blocking, low priority customers leave the system and retry until they find the server free. In related bibliography ([6], [12], the high priority customers have either preemptive or non-preemptive priority over the low priority customers.

There has been a very rich literature on queues with

*Acknowledgment: This research is partially supported by the Hungarian Science and Technology Foundation, Hungarian-French Bilateral Cooperation under grant TéT 10-1-2011-0741, FR-25/2010.

vacations. For some basic results see, for example [7], [10].

Using wireless sensor networks, see [1], [11] one of the biggest problem is the lifetime of the sensor which depends of the battery life. Most of the time it is very hard to change, or it is impossible to replace or recharge the batteries of the sensors. Because these facts, the battery resources is very important. That means that the power consumption of the sensor networks is one of the most important property. The lifetime of the sensor determine the lifetime of the network too. Of course the lifetime of the sensors will be longer if power consumption is reduced. A major approach to reduce the power consumption of mobile nodes is to use vacation or sleep periods when the nodes/servers are awake. Current standards of mobile communication such as WiFi, 3G and WiMAX have provisions to operate the mobile node in power save mode in case of low uses scenarios.

In the present paper we introduce a new model which combines the components of finite-source priority queues with a single server under multiple vacations. To the best knowledge of the authors, there is no such model in the literature. It is primary based on [7], and [14]. Because of the fact, that the state space of the describing Markov chain is very large, it is rather difficult to calculate the system measures in the traditional way of writing down and solving the underlying steady-state equations. To simplify this procedure we used the software tool MOSEL (Modeling, Specification and Evaluation Language), see [4], to formulate the model and to obtain the performance measures. By the help of MOSEL we can use various performance tools (like SPNP Stochastic Petri Net Package) to get these characteristics. The results of the tool can graphically be displayed using IGL (Intermediate Graphical Language) which belongs to MOSEL.

The organization of the paper is as follows. Section 2 contains the corresponding queueing model with components (finite sources, the orbit, the queue and service area) to study the behavior of the sensor nodes and the derivation of the main steady-state performance measures. In Section 3, we present some numerical examples and the results are graphically displayed using the IGL (Intermediate Graphical Language) interpreter which belongs to MOSEL. By the help of these figures we illustrate the effect of the arrival rate and the listening period on the mean response times in the queue and in the orbit, respectively.

## II. SYSTEM MODEL

Let us consider a single server queue with a finite user population that consists $K$ customers. Each customer generates a new request from the source according to an exponentially distributed time with parameter $\lambda^*$. We define two priorities of requests ($P_i,\ i = 1, 2$). The probability that a new generated request has $P_1$ priority is $p_1$ and $P_2$ priority is $p_2$. That means, requests from customers with $P_i$ priority arrive according to an exponentially distributed time with parameter $\lambda_i = p_i\lambda^*$. The $P_1$ requests go directly to a FIFO queue waiting to be served. If a $P_2$ request finds the server in busy or unavailable state, goes to the orbit. These requests waiting in the orbit retry to find the server idle and available state according to a Poisson flow with retrial rate $\nu$. We assume that $P_1$ requests have non-preemptive priority over $P_2$ requests. The service times for each request are exponentially distributed with parameter $\mu$. The server could be is two states:

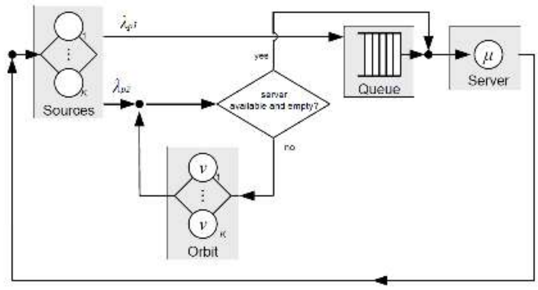


Figure 1: A retrial queue with components.

- *available state:* If the server is in available state it could start serving the arriving requests.
- *sleeping state:* If the server is in sleeping state the server does not work, so every new $P_1$ request goes to the queue and $P_2$ requests go to the orbit. During the sleeping period the server could not serve any request.

The server is busy when the server is in available state and at least one request is in the service area. The server is idle when the server is in available state and there is no request in the service area.

The server starts with a listening period. The time of this listening period is assumed to be exponentially distributed with parameter $\alpha$. If no customer arrives during this period, the server will enter into the sleeping state. The time of the sleeping period is exponentially distributed with parameter $\beta$.

When the sleeping period is terminated, then the node wakes up. If there is a $P_1$ priority request waiting in the queue the server begins to serve it. In the opposite case, when there is no $P_1$ priority request waiting in the queue, the server remains in the available state, it will start a listening period. Until the listening period finished, the $P_1$ and $P_2$ requests arriving from the source or from the orbit could access the server. If the listening period expires without any arrivals the node will enter into the sleeping mode.

The operational dynamics of the system can be seen in the corresponding queueing model see Fig. 1.

We introduce the following notations (see the summary of the model parameters in Table 1)

- $k(t)$ is the number of customers in the source at time $t$,
- $q(t)$ denote the number of $P_1$ requests in the queue at time $t$,
- $o(t)$ is the number of $P_2$ requests in the orbit at time $t$.
- $y(t) = 0$ if the server is available and $y(t) = 1$ if the server is in sleeping period at time $t$
- $c(t) = 0$ if the server idle and $c(t) = 1$ if the server is busy at time $t$

It is easy to see that $k(t) = K - q(t) - o(t) - c(t)$.

Table 1: Overview of model parameters.

| Parameter | Maximum | Value at $t$ |
|---|---|---|
| Number of sources | $K$ (population size) | $k(t)$ |
| $P_1$ generation rate | | $\lambda_{p1}$ |
| $P_2$ generation rate | | $\lambda_{p2}$ |
| Total gen. rate | $\lambda_{p1}K + \lambda_{p2}K$ | $(\lambda_{p1} + \lambda_{p2})k(t)$ |
| Requests in queue | $K$ | $q(t)$ |
| Service rate | | $\mu$ |
| Busy servers | 1 (number of servers) | $c(t)$ |
| Cust. in service area | $K$ | $c(t)+q(t)$ |
| Requests in Orbit | $K$ (orbit size) | $o(t)$ |
| Retrial rate | | $\nu$ |

To preserve mathematical tractability, all inter-event times (i.e., request generation time, service time, retrial time, listening time, vacation time) involved in the model contruction are assumed to be exponentially distributed and independent of each other. The system state at time $t$ can be described by a four-dimensional Continuous Time Markov Chain (CTMC):

$$X(t) = (y(t); c(t); q(t); o(t)).$$

The steady state probabilities are denoted by

$$p_{y,c,q,o} = \lim_{t\to\infty} P(y(t) = y, c(t) = c, q(t) = q, o(t) = o),$$

where $y = 0, 1$ and $c = 0, 1$.

Note that the stationary distribution always exists because the state space of the CTMC is finite.
In this paper, the MOSEL-tool is used to compute the stationary distribution and main the performance measures. Because of the page limitation the derivation of the performance measures are omitted, since it is traditional if we know the steady-state distribution of the system. For an easier understanding, see for example [5], [14].

Once we have obtained the above defined probabilities, the main steady-state system performance measures as usual can be derived in the following way:

- *Utilization of the server*

$$U_S = \sum_{q=0}^{K-1}\sum_{o=0}^{K-q} P(0, 1, q, o)$$

- *Availability of the server*

$$A_S = \sum_{c=0}^{1}\sum_{q=0}^{K-c}\sum_{o=0}^{K-q} P(0, c, q, o)$$

- *Mean number of requests at the orbit*

$$\overline{O} = E(o(t)) = \\ = \sum_{y=0}^{1}\sum_{c=0}^{1}\sum_{q=0}^{K-c}\sum_{o=0}^{K-c-q} qP(y, c, q, o)$$

- *Mean number of requests in the queue*

$$\overline{Q} = E(q(t)) = \\ = \sum_{y=0}^{1}\sum_{c=0}^{1}\sum_{q=0}^{K-c}\sum_{o=0}^{K-c-q} oP(y, c, q, o)$$

- *Mean number of requests in the orbit or in the queue or in service*

$$\overline{M} = E(o(t) + q(t) + c(t)) = \\ = \overline{O} + \overline{Q} + \sum_{q=0}^{K}\sum_{o=0}^{K-q} P(0, 1, q, o)$$

- *Mean number of active sources*

$$\overline{\Lambda} = K - \overline{M}$$

- *Mean overall generation rate:*

$$\overline{\lambda} = (\lambda_{p1} + \lambda_{p2})\overline{\Lambda}$$

- *Mean time spent in queue:*

$$ET_q = \frac{\overline{Q}}{\overline{\lambda}}$$

- *Mean time spent in orbit:*

$$ET_o = \frac{\overline{O}}{\overline{\lambda}}$$

## III. NUMERICAL RESULTS

In this section, we present some numerical results in order to illustrate graphically the efficiency of vacation in sensor networks. For the numerical explorations the corresponding parameters of Dimitriou [7] are used. In the future research our plan is to change these parameters for other situations. Further parameters are summarized in Table 2.

Several numerical experiments have been carried out to examine the performance behavior of the model with respect to various parameter values.

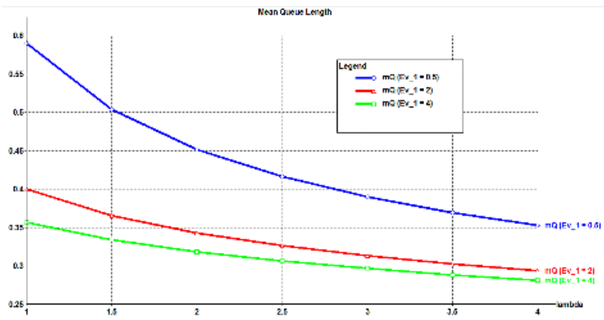


Figure 2: Mean queue length vs $\lambda$ .

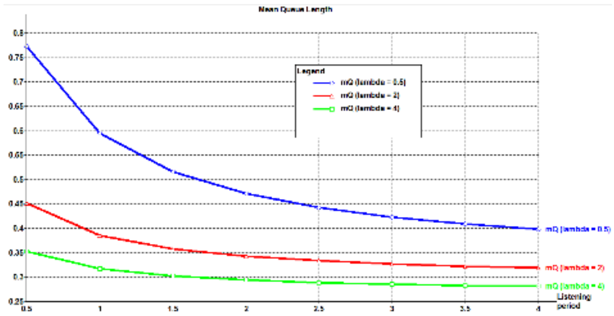


Figure 6: Mean queue length vs listening period .

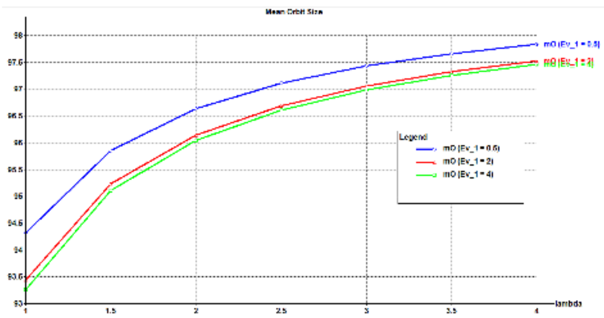


Figure 3: Mean orbit size vs $\lambda$ .

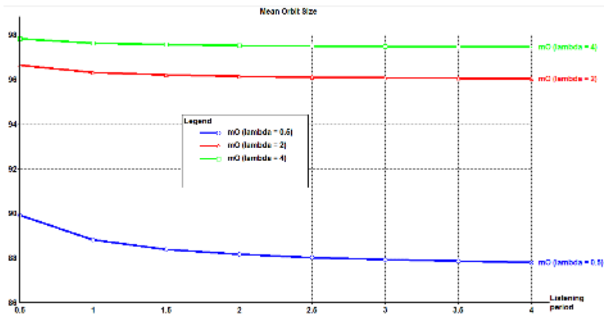


Figure 7: Mean orbit size vs listening period .

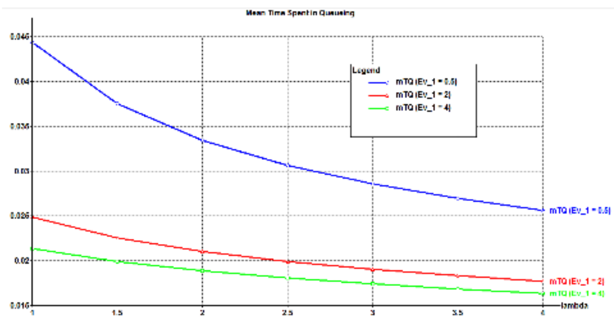


Figure 4: Mean time spent in queue vs $\lambda$ .

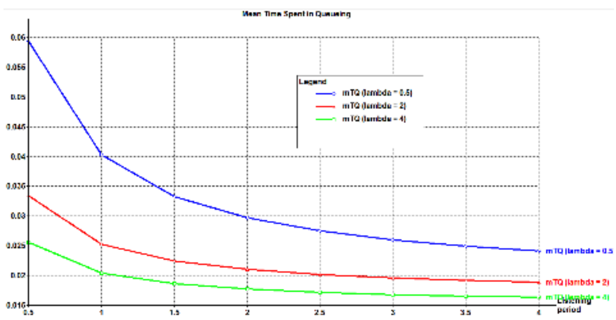


Figure 8: Mean time spent in queue vs listening period .

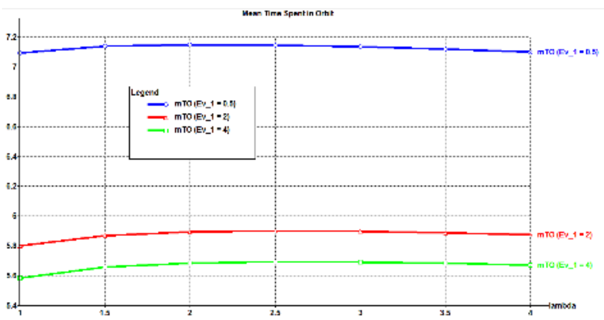


Figure 5: Mean time spent in orbit vs $\lambda$.

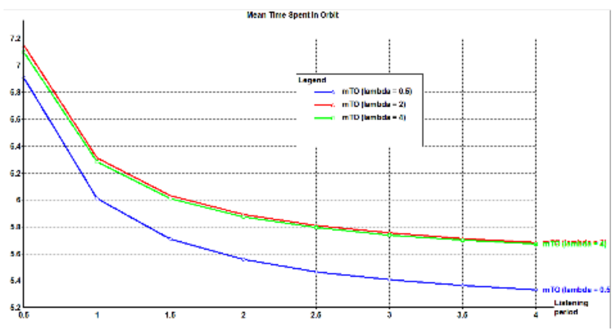


Figure 9: Mean time spent in orbit vs listening period.

- In Figure 2 we can see the mean queue length as a function of the $P_1$ customer generation rate. As we see the mean queue length is decreasing when the $P_1$ generation rate is increasing. We can observe that a lower listening period $Ev_1 = 0.5$ results a higher value of mean queue length. That means, using larger listening period $Ev_1 = 4$ we get lower mean queue length for all value of $\lambda$

- In Figure 3 we can see the mean orbit size as a function of the $P_1$ customer generation rate ($\lambda$). As we see the mean orbit size will be greater as we increase the $\lambda$. Because the mean orbit size is between 93 and 98, the mean number of active customers in the source is between 2 and 7. Therefore, the $P_1$ requests generation rate is between $2\lambda$ and $7\lambda$. Investigating these values we can understand the difference between the Figure 2. and Figure 3.

- In Figure 4-5 we investigate the effect of requests generation rate on the mean time spent is queue and in orbit. As we see, the mean time spent in queue will be smaller using higher generation rate. We can understand these property if we take into account that a higher generation rate results more requests in the orbit. The total generation rate, namely $(\lambda_1 + \lambda_2)k(t)$ will be smaller, because $k(t)$ will be smaller, which results a lower mean response time for requests in the queue.

- In Figure 6 the effect of the listening period is demonstrated on the mean queue length, in case when $\lambda = 0.5, 2, 4$. As we can see, a low value for mean listening period results in a higher value for mean queue length. Increasing the listening period the size of the mean queue length decreases. We can observe, that using higher generation rates ($\lambda$) we get lower value for mean queue length.

- In Figure 7 the mean orbit size is shown as a function of listening period. As we could see, a higher generation rate results in a higher values for mean orbit size. Because the mean orbit size is very high (between 88 and 98) according to the population size ($K = 100$), the mean number of active customers in the source is very low (between 2 and 12). Therefore, a higher generation rate results in a lower value for the mean queue length.

- In Figure 8-9 we depict the mean time spent in queue and in orbit as a function of the listening period. In both case we see, that using higher listening period we get lower mean time spent is queue and in orbit.

Table 2: Numerical values of model parameters.

| Parameter | Symbol | Value |
|---|---|---|
| $P_1$ generation rate | $\lambda_1$=$\lambda$ | [1,4] |
| $P_2$ generation rate | $\lambda_2$=2 $\lambda$ | [1,4] |
| Number of sources | $K$ | 100 |
| Retrial rate | $\nu$ | 0.8 |
| Service rate | $\mu$ | 20 |
| Mean time of sleeping period | $Ev_0$ | 2 |
| Mean time of listening period | $Ev_1$ | $0.5, 2, 4$ |

## IV. CONCUSIONS

In this paper we have proposed a new single server finite-source retrial queueing model with non preemptive priorities and repeated vacations with possible application to sensor networks. The impact of parameters on the main performance of the system has been investigated by using the modeling tool MOSEL. To our best knowledge, this is the first proposal for the use of priority finite-source retrial queues with state dependent vacation times to model sensor networks. In our future work we would like to investigate more complex sensor models concentrating on power consumption problems, too.

The authors are very grateful to the referees for their helpful remarks and comments which have improved the presentation.

# References

[1] Akyildiz, I. F., and Vuran, M. C. *Wireless Sensor Networks*. John Wiley & Sons, July 2010.

[2] Artalejo, J., and Gomez-Corral, A. *Retrial Queueing Systems: A Computational Approach*. Springer Verlag, 2008.

[3] Artalejo, J. R. Accessible bibliography on retrial queues: Progress in 2000-2009. *Mathematical and Computer Modelling 51*, 9-10 (2010), 1071–1081.

[4] Begain, K., Bolch, G., and Herold, H. *Practical performance modeling, application of the MOSEL language*. Kluwer Academic Publisher, Boston, 2001.

[5] Bolch, G., Greiner, S., de Meer, H., and Trivedi, K. *Queueing Networks and Markov Chains*, 2nd ed. John Wiley & Sons, New York, 2006.

[6] Choi B.D, C. Single server retrial queues with priority customers. *Mathematical and Computer Modelling 30* (1999), 7–32.

[7] DIMITRIOU, I. Analysis of a priority retrial queue with dependent vacation scheme and application to power saving in wireless communication systems. *The Computer Journal 55* (2012).

[8] DO, T. V. An efficient computation algorithm for a multiserver feedback retrial queue with a large queueing capacity. *Applied Mathematical Modelling 34* (2010), 2272–2278.

[9] DO, T. V. An efficient solution to a retrial queue for the performability evaluation of DHCP. *Computers & OR 37*, 7 (2010), 1191–1198.

[10] DOSHI., B. Queueing system with vacations-a survey. *Queueing Systems 1* (1986), 29–66.

[11] DRESSLER, F. *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons, 2007.

[12] FALIN, G., AND ARTALEJO, J. On the single server retrial queue with priority customers. *Queueing Systems 14* (1993), 439–455.

[13] FALIN, G., AND TEMPLETON, J. *Retrial Queues*. Chapman & Hall, 1997.

[14] WCHNER, P., SZTRIK, J., AND DE MEER, H. Modeling wireless sensor networks using finite-source retrial queues with unreliable orbit. *Springer Lecture Notes in Computer Science 6821* (2011), 275–285.