A multi-server batch arrival retrial system

Andreas Brandt[brandt@wiwi.hu−berlin.de]
Wirtschaftswissenschaftliche Fakultät,
Humboldt-Universität zu Berlin,
Spandauer Str. 1, D-10178 Berlin, Germany

Torsten Schiemann[schieman@informatik.hu−berlin.de]
Mathematisch-Naturwissenschaftliche Fakultät II,
Humboldt-Universität zu Berlin,
Hausvogteiplatz 5-7, D-10117 Berlin, Germany

Abstract

The following retrial system is considered: at a group of C servers batches of requests arrive from outside according to a Poisson process. The batches are of size d_i with probability $p_i, i = 1, 2$. If an arriving request finds a server free it immediately receives service, otherwise it enters orbit, and after a constant time τ the request retrials from orbit to the servers for getting service now. If a server is free then it gets service, otherwise it returns to orbit again and retrials later in the same way (potentially infinitly often) until it gets service, provided the system is stable. The service times are of the form $s + Y$, where $s \in \mathbb{R}_+$ and Y is an exponentially distributed r.v. The described retrial system models the PPP connection for a pool of modems. Of interest are the fraction of all requests from outside and going to orbit and the mean and variance of the sojourn time of a batch of size d_i in the system, defined as the time from the arrival of a batch until all requests of this batch have received their service. Besides the stability condition we consider two limiting cases of the model for which analytical results are given and a third limiting case for which an approximation is proposed. Since even an approximate numerical computation of the performance measures would lead to too complex algorithms (in time and space) a simulation program for the system has been written. The presented simulation results illustrate the impact of the system parameters.

Mathematics Subject Classification (MSC 2000): 60K25, 68M20, 65C05.

Keywords: many-server retrial system; batch arrivals; constant retrial times; PPP connection; simulation results.

1 Introduction and model description

In this paper we consider a multi-server batch arrival retrial system, modelling the PPP connection for a pool of modems. The model is as follows, cf. Figure 1.1. At a group of C servers, from outside there arrives a Poisson stream $\Phi = \{T_\ell\}_{\ell=1}^{\infty}$, $0 < T_1 < T_2 < \ldots$, of intensity λ of batches of requests. Let $G_\ell, \ell = 1, 2, \ldots$ denote the size of the batch arriving at T_ℓ. We assume that Φ and $\{G_\ell\}_{\ell=1}^{\infty}$ are stochastically independent and that $\{G_\ell\}_{\ell=1}^{\infty}$ is a sequence of i.i.d. r.v.'s. Let G be a generic r.v. and $g_n := P(G = n)$, $n \in \mathbb{Z}_+ \setminus \{0\}$ the batch size distribution with mean $m_G := EG$.

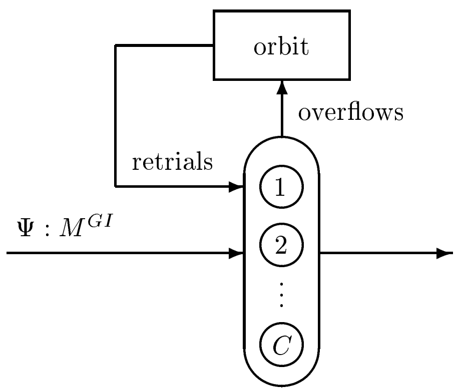

Figure 1.1: *Multi-server batch arrival retrial system*

The stream $\Psi = \{[T_\ell, G_\ell]\}_{\ell=1}^{\infty}$ of requests from outside (external input) is denoted by the symbol M^{GI}, where in case of single arrivals, i.e. if $G_\ell = 1$, $\ell = 1, 2, \ldots$, the stream of requests is already given by Φ. In this case we identify Φ with the stream of requests from outside and use the symbol M for the arrival process. If an arriving request from outside (external arrival) finds some server free it immediately occupies the server and leaves the system after service. Otherwise, if all servers are busy, the request enters orbit and produces a source of repeated calls.* More precisely, after a random time, whose generic random variable is denoted by R with distribution $R(t) := P(R \leq t)$ and mean $m_R := ER$, the request retrials, i.e., it returns to the C servers. If a server is free then it gets service, otherwise it returns to orbit again and retrials later in the same way. The retrial times are assumed to be independent, i.e., the requests in orbit behave independent, and there is no

*The dynamics imply that the overflow process of the batch arrivals from outside is a batch process, too.

limit for repeated attempts. Further, the capacity of orbit is unlimited. The service times of the requests are i.i.d. r.v.'s, where S denotes a generic r.v. Denote by $B(t) := P(S \leq t)$ its distribution function, by $m_B := ES$ its mean and by $\sigma_B^2 := D^2 S$ and $c_B^2 := \sigma_B^2/(m_B)^2$ its variance and squared coefficient of variation, respectively. We abbreviate the system as $M^{GI}/GI/C/0$ with $R(t)$ distributed retrial times. The offered load ϱ to the system is given by

$$\varrho = \lambda m_G m_B. \tag{1.1}$$

By means of the model assumptions and work conservation arguments it can be shown that

$$\varrho < C \tag{1.2}$$

is the stability condition of the system, which is assumed to be fulfilled in the following. Further, the work conservation principle implies that the utilization of the servers u, i.e., the fraction of time during which the servers are busy, is given by

$$u = \varrho/C. \tag{1.3}$$

Motivated by the application mentioned above, we assume in the following:

(A1) $R(t) = \mathbb{1}\{t \geq \tau\}$, where τ is a positive parameter, i.e., the retrial times are constant τ.

(A2) $S(t) = \mathbb{1}\{t \geq s\}(1 - e^{-\mu(t-s)})$, where $s \in \mathbb{R}_+$ and $\mu \in \mathbb{R}_+ \setminus \{0\}$, i.e., the service time S is of the form $S = s + Y$ and Y is an exponentially distributed r.v. with parameter μ.

(A3) $g_n = p_1 \mathbb{1}\{n = d_1\} + p_2 \mathbb{1}\{n = d_2\}$, $n \in \mathbb{Z}_+ \setminus \{0\}$, where $d_1, d_2 \in \mathbb{Z}_+ \setminus \{0\}$, $d_1 \neq d_2$, $p_1 \in [0,1]$ and $p_2 = 1 - p_1$, i.e., the batch sizes take the values d_i with probability p_i, $i = 1,2$ or equivalently $p_1 = P(G = d_1) = 1 - P(G = d_2)$.

Remark 1.1 *Assumption (A1) implies that the stream of retrial requests is a batch arrival process, too. In view of (A3), we call the batches of size d_i also type i-batches, $i = 1,2$.*

The assumptions (A2) and (A3) imply

$$\begin{aligned} m_B &= s + \frac{1}{\mu}, \quad \sigma_B^2 = \frac{1}{\mu^2}, \quad c_B^2 = (\frac{1}{1+s\mu})^2, & (1.4) \\ m_G &= d_1 p_1 + d_2 p_2, & (1.5) \end{aligned}$$

and the stability condition (1.2) reads

$$\lambda(d_1 p_1 + d_2 p_2)(s + 1/\mu) < C. \tag{1.6}$$

Let $V_i,\ i=1,2$ be the stationary sojourn time of an arriving batch of size d_i from outside at the retrial systems, i.e., the time from the arrival of the type i-batch until all requests of the batch have received their service. More precisely, if $V_i^1,\ldots,V_i^{d_i}$ denote the individual sojourn times of the requests of the batch, then $V_i=\max(V_i^1,\ldots,V_i^{d_i})$. We are interested in the means $m_{V,i}:=EV_i$ and variances $\sigma^2_{V,i}:=D^2V_i$ or equivalently in the squared coefficients of variation $c^2_{V,i}:=\sigma^2_{V,i}/(m_{V,i})^2,\ i=1,2$. For the corresponding overall stationary sojourn time V of a typical arriving batch we find for the mean $m_V:=EV$ and second moment $m_V^{(2)}:=EV^2$

$$m_V=p_1m_{V,1}+p_2m_{V,2},\quad m_V^{(2)}=p_1m_{V,1}^{(2)}+p_2m_{V,2}^{(2)} \tag{1.7}$$

in view of (A3), where $m_{V,i}^{(2)}:=EV_i^2,\ i=1,2$ are the second moments of the V_i. For the squared coefficient of variation $c_V^2=m_V^{(2)}/(m_V)^2-1$ and for the variance $\sigma_V^2:=D^2V$ from (1.7) and $m_{V,i}^{(2)}=(m_{V,i})^2(c^2_{V,i}+1)$ we find that

$$c_V^2=\frac{1}{(m_V)^2}\sum_{i=1}^{2}p_i(m_{V,i})^2(c^2_{V,i}+1)-1,\quad \sigma_V^2=(m_V)^2c_V^2. \tag{1.8}$$

Besides the sojourn time characteristics we are interested in the fraction p_B of all requests from outside and which go to orbit, i.e., which have to retrial for getting service.

There is a huge literature on retrial systems, where most of the papers deal with single server retrial systems. Surveys on retrial systems can be found e.g. in [AR1], [AR2], [F], [FT], [KL], [YT1], [YT2]. Different mathematical methods and numerical algorithms where applied, cf. the mentioned survey papers and also e.g. [FG], [G], [NR], [S1], [S2], [W]. Some more recently papers are e.g. [BDK], [CCD], [CR], [HLZ], [WCL]. However, the most relevant paper to ours seems to be [HLZ], dealing with the $BMAP/PH/C/K$ system with PH retrial times, where $BMAP$ stands for a Batch Markovian Arrival Process, PH for phase-type distributed service and retrial times, respectively, and K for the number of waiting places in front of the C servers. Since a M^{GI} process is a special $BMAP$ and GI service times as well as GI retrial times can be approximated by a PH distribution, our particular $M^{GI}/GI/C/0$ system with GI retrial times τ can be approximated by a $BMAP/PH/C/K$ system with $K=0$ and PH retrial times. But $[HLZ]$ deals only with the stability condition for the $BMAP/PH/C/K$ system with PH retrial times and not with the problem of calculating the stationary state distribution or stationary batch sojourn times. For our best knowledge a numerical or analytical treatment of our model seems not to be available from the literature. Also, developing algorithms for computing the performance measures of our model using phase-type approximations would result in very complex algorithms which seem not to be the appropriate approach. Thus we decided to simulate the model. Special cases which can be treated analytically offer the possiblity for checking our simulation program. The paper is organized as follows. In Section 2 we present three special cases which are useful for checking the accuracy of our simulation program. In Section 3 we describe the implementation of the simulation program and give several numerical and simulation results. The parameters are chosen according to the modelling of a real life modem pool model, cf. Section 1. The results demonstrate the strong impact of the various parameters.

2 Analytically tractable special cases

For checking the accuracy of our simulation program as well as getting informations concerning the number of events which have to be simulated in order to obtain sufficiently correct statistics, we are interested in analytical results. In the following we consider three limiting cases, where for the first two analytical results are available and for the third one an approximation is given.

2.1 Limiting case $C \to \infty$

For $C \to \infty$ the dynamics of the retrial system converge to those of a $M^{GI}/GI/\infty$ system, where the batch sizes and service times are distributed according to the assumptions (A3) and (A2), respectively. Since in a $M^{GI}/GI/\infty$ system all requests will be accepted at their arrival, there are no retrials, and the system is always stable. Consider a type i-batch of size $d_i \in \mathbb{Z}_+ \setminus \{0\}$, $i = 1, 2$ arriving at the infinite server system and let $S_j = s + Y_j$, $j = 1, \ldots, d_i$ be its individual service times, where the Y_j, $j = 1, \ldots, d_i$ are i.i.d. exponentially distributed r.v.' with parameter μ. Since in an infinite server system the service of all requests starts immediately at their arrival, the individual sojourn times of requests are just their service times. Thus the sojourn time V_i of the arriving type i-batch is given by

$$V_i = \max(S_1, \ldots, S_{d_i}) = s + \max(Y_1, \ldots, Y_{d_i}), \tag{2.1}$$

cf. also Section 1. From (2.1) for $i = 1, 2$ it follows

$$m_{V,i} = s + \frac{1}{\mu}\sum_{j=1}^{d_i}\frac{1}{j}, \quad \sigma^2_{V,i} = \frac{1}{\mu^2}\sum_{j=1}^{d_i}\frac{1}{j^2}, \tag{2.2}$$

$$c^2_{V,i} = \frac{\sigma^2_{V,i}}{(m_{V,i})^2}. \tag{2.3}$$

The overall sojourn time characteristics are then given by (1.7), (1.8). Further, it holds

$$p_B = 0. \tag{2.4}$$

2.2 Limiting case $d_1 = 1, p_1 = 1, s = 0, \tau \to 0$

If $d_1 = 1$, $p_1 = 1$ and $s = 0$ then $p_2 = 0$ and the arrival stream Φ of requests from outside is a Poisson stream (of intensity λ) of single arrivals and the service times are exponentially distributed with mean $1/\mu$. Since for $\tau \to 0$ the requests in orbit check immediately whether a server is free, the dynamics of the retrial system converge to those of a $M/M/C/\infty$ system (with parameters λ, μ). The stability condition (1.2) reads $\varrho < C$, where $\varrho = \lambda/\mu$. Denote by X the stationary number of requests in the $M/M/C/\infty$ system, which corresponds to the cumulative number of requests in the C servers and in orbit. The stationary distribution $p_n := P(X = n), n \in \mathbb{Z}_+$ is given by, cf. e.g. [GH],

$$p_n = \begin{cases} \dfrac{\varrho^n}{n!} p_0, & n = 0, \dots, C-1 \\ \dfrac{\varrho^n}{C! C^{n-C}} p_0, & n = C, C+1, \dots \end{cases} \tag{2.5}$$

and

$$p_0 = \left(\sum_{i=0}^{C-1} \frac{\varrho^i}{i!} + \frac{\varrho^C}{C!(1 - \varrho/C)} \right)^{-1}. \tag{2.6}$$

The blocking probability in this limiting case is given by

$$p_B = \sum_{n=C}^{\infty} p_n = 1 - p_0 \sum_{n=0}^{C-1} \frac{\varrho^n}{n!}. \tag{2.7}$$

For the mean sojourn time m_V from Little´s formula it follows

$$m_V = \frac{1}{\lambda} EX, \tag{2.8}$$

where for the mean number EX of requests it holds, cf. e.g. [GH],

$$EX = \frac{\varrho^C \varrho}{C! C (1 - \varrho/C)^2} p_0 + \varrho. \tag{2.9}$$

Remark 2.1 *By means of continuous time Markov chains, the limiting case $s = 0, \tau \to 0$ could be treated for the general batch arrival stream given by assumption (A3). However, the theoretical outline and numerical algorithms are more complex and will not be given here.*

2.3 Approximation for the limiting case $d_1 = 1, p_1 = 1, \tau \to \infty$

As in the previous limiting case let the arrival stream of requests from outside be a Poisson stream Φ (of intensity λ) of single arrivals, i.e. $d_1 = 1, p_1 = 1$. The stability condition reads as $\varrho = \lambda m_B < C$, and the stream of retrial requests from orbit consists of single arrivals, too. Denoting by $\Phi^{(r)} = \left\{T_\ell^{(r)}\right\}_{l=1}^{\infty}$, $0 < T_1^{(r)} < T_2^{(r)} < \ldots$, this stream of requests from orbit, for $\tau \to \infty$ the processes Φ and $\Phi^{(r)}$ tend to become stochastically independent. Although $\Phi^{(r)}$ does not converge to a Poisson process as $\tau \to \infty$, we approximate $\Phi^{(r)}$ by a Poisson process of intensity $\lambda^{(r)}$ in the following. Under this assumption the cumulative arrival process arriving at the C servers is a Poisson process $\tilde{\Phi}$ of intensity $\tilde{\lambda} := \lambda + \lambda^{(r)}$, and the dynamics of the C servers correspond to those of a $M/GI/C/0$ system. The steady state distribution p_n, $n = 0, \ldots, C$, of the number of busy servers is given by

$$p_n = p_0 \frac{\tilde{\varrho}^n}{n!}, \quad n = 0, \ldots, C, \tag{2.10}$$

$$p_0 = \left(\sum_{n=0}^{C} \frac{\tilde{\varrho}^n}{n!} \right)^{-1}, \tag{2.11}$$

where

$$\tilde{\varrho} = \tilde{\lambda} m_B = \varrho + \varrho^{(r)}, \quad \varrho^{(r)} = \lambda^{(r)} m_B. \tag{2.12}$$

The intensity of the process of rejected requests from the C servers (overflow process) is given by

$$\lambda^{(r)} = \tilde{\lambda} p_C,$$

or, equivalently, in terms of the retrial load $\varrho^{(r)} = \tilde{\varrho} p_C$. The PASTA-property implies that the blocking probability p_B of an arriving request from outside is given by

$$p_B = p_C. \tag{2.13}$$

For computing p_C we have to determine $\tilde{\lambda}$ or equivalently $\varrho^{(r)}$, cf. (2.12). From (2.10)-(2.13) it follows that $\varrho^{(r)}$ is given by the solution of the fixed point equation

$$\varrho^{(r)} = f(\varrho^{(r)}), \tag{2.14}$$

where

$$f(\varrho^{(r)}) := (\varrho + \varrho^{(r)}) \left(\sum_{n=0}^{C} \frac{(\varrho + \varrho^{(r)})^n}{n!} \right)^{-1} \frac{(\varrho + \varrho^{(r)})^C}{C!} \tag{2.15}$$

is a non-linear function. Eq. (2.14) can be solved by using the iteration $\varrho_{\ell+1}^{(r)} = f(\varrho_\ell^{(r)})$, $\ell = 0, 1, \ldots$ with starting value $\varrho_0^{(r)} := 0$.

Remark 2.2 *Since overflow processes usually have a peakedness $Z > 1$, cf. e.g. [W], and peaked streams of requests usually lead to larger blocking probabilities than a Poisson process with the same mean, our proposed approximation should underestimate the exact blocking probability, cf. also the simulation results in Section 3.*

3 Implementation and simulation results

The simulation of the described model, cf. Section 1, and the computation of the three special cases, cf. Sections 2.1 – 2.3, are implemented in the program simula.exe. The source code is written in C++ for Windows 9x, Windows NT or Windows XP and has been compiled by Visual C++ 6.0. The program provides a graphical user interface. After starting the program the user will be requested for the input of the following parameters, described in Section 1 with the exception of N, where the standard configuration, given in (3.1) and (3.2), is suggested:

Number of served batches of requests	–	N,
Intensity of batch arrivals	–	λ,
Batch size	–	d_1, d_2,
Probability of the batch size d_1, d_2	–	p_1, p_2,
Number of servers	–	C,
Constant part of the service time	–	s,
Mean of the exponential part of the service time	–	$\frac{1}{\mu}$,
Mean retrial time	–	τ.

The user can choose between a constant retrial time, i.e. $R \equiv \tau \equiv m_R$, or a random retrial time R with a uniform distribution on $[\tau/2, 3\tau/2]$, where $\tau = m_R$. The number N denotes the number of served batches of requests that will be simulated, provided the program run is stable. For the parameters it is assumed

$$N \in \mathbb{N}, \quad \lambda > 0, \quad d_1, d_2 \in \mathbb{N}, \quad p_1 = 1 - p_2 \in [0,1],$$
$$C \in \{1, 2, \ldots, 1000\}, \quad s, 1/\mu \geq 0, \quad \tau > 0.$$

After input of the parameters and starting the program (please press the start bottom) first the stability condition (1.2), (1.6) will be checked before the simulation is running. After finishing the simulation of N batches of requests and computation of the three limiting cases the following output is written to the screen:

Mean sojourn time	–	m_V,
Mean sojourn time for type i-batches	–	$m_{V,i}$,
Variance of the sojourn time	–	σ_V^2,
Variance of the sojourn time for type i-batches	–	$\sigma_{V,i}^2$,
Squared coefficient of variation	–	c_V^2,
Squared coefficient of variation for type i-batches	–	$c_{V,i}^2$,
Blocking probability	–	p_B.

Before starting the simulation the user is requested for a file name, where the data have to be stored. After finishing the simulation the statistical data as well as the input data are stored in a LaTeX table. Note that in case of $u = \varrho/C \gtrapprox 0,95$ and/or $\tau = m_R$ large, the simulation may become very slowly and/or instable in view of fast increasing memory and/or computing time requirements. Tables 3.1 - 3.7 have been drawn up by using the program simula.exe. If not stated otherwise, we used the following parameters, which are realistic choices for modeling the PPP connection for a pool of modems [S]:

$$d_1 = 1,\ d_2 = 20,\ p_1 = p_2 = 0.5, \tag{3.1}$$

$$C = 120,\ s = 60\ [sec],\ \tau = m_R = 60\ [sec]. \tag{3.2}$$

The range for $1/\mu$ (mean of the exponential part of the service time) and the server utilization $u = \varrho/C$, cf. (1.1), (1.3) - (1.5), which is of practical relevance, is

$$1 \leq 1/\mu \leq 10, \quad 0.5 \leq u \leq 0.9. \tag{3.3}$$

(Note, concerning C also $C \in \{16, 32\}$ is of interest).

Tables 3.1 and 3.2 illustrate the convergence of the overall performance measures p_B, m_V and c_V^2 to the steady state limits as the number N of simulated served batches increases and suggest/give hints for realistic values of N to be chosen for obtaining statistically reliable simulation results. Thus we used $N = 10^7$ for lower utilizations and $N = 2.5 \cdot 10^8$ for higher utilizations in our simulation runs. In Tables 3.3 - 3.5 there are presented results for the limiting cases given in Sections 2.1 - 2.3. More precisely, in Table 3.3 there are presented the analytical results for the limiting case $C \to \infty$ (cf. Section 2.1) and the corresponding simulated values for $C = 120$, which can be considered as a good approximation for the infinite server system in view of $\varrho = 1.2$. The simulation results confirm this consideration. In Table 3.4 there are presented results for the limiting case given in Section 2.2. The values in the row for $\tau = 0$ are computed numerically. The simulated values show its convergence for $\tau \to 0$, as expected. The proposed approximation given in Section 2.3 for the limiting case $d_1 = 1, p_1 = 1$ and $\tau \to \infty$ is illustrated in Table 3.5. In the row $\tau = \infty$ there is given the – numerically computed – proposed approximation for p_B. As seen from the simulation results this approximation is indeed a lower bound for p_B, as heuristically expected from general experiences with overflow processes, cf. Remark 2.1. In Table 3.6 there are given the blocking probabilities of the requests and sojourn time characteristics for batches for a model of a pool of modems whith its PPP connections [S]. The simulation results illustrate that the mean sojourn time $m_{V,i}$ of a type i-batch strongly depends on its batchsize $d_i, i = 1, 2$, cf. (3.1). Also if u increases the $m_{V,i}$ increase considerably, compared to the mean service time $m_B = s + 1/\mu$ of a request, cf. (1.4). The retrial times are realized in a modem pool architecture by a timer of constant time $R \equiv m_R = \tau$. It is an interesting question whether an random retrial time with the same mean would provide better performance measures. In Table 3.7 there are given p_B, m_V and c_V^2 for constant and uniformly distributed – over the interval $[30, 90]$ – retrial times of mean $m_R = 60$. The presented results give a mixed picture. Under some parameter constellations there is an improvement of the performance measures, whereas in other cases they become worse. Thus there is no clear answer whether a random timer should be chosen.

N	p_B	m_V	c_V^2
100000	0.06234	67.25560	0.0716
200000	0.06381	67.38008	0.0740
300000	0.06402	67.40065	0.0745
400000	0.06428	67.42748	0.0746
500000	0.06454	67.43740	0.0745
600000	0.06481	67.45248	0.0747
700000	0.06479	67.46547	0.0752
800000	0.06490	67.46263	0.0751
900000	0.06551	67.50247	0.0757
1000000	0.06555	67.50460	0.0758
$\vdots$	$\vdots$	$\vdots$	$\vdots$
4400000	0.06546	67.50654	0.0762
4500000	0.06550	67.50872	0.0763
4600000	0.06549	67.50818	0.0762
4700000	0.06548	67.50569	0.0762
4800000	0.06547	67.50304	0.0761
4900000	0.06551	67.50756	0.0762
5000000	0.06548	67.50604	0.0762
5100000	0.06547	67.50494	0.0762
5200000	0.06539	67.50151	0.0761
5300000	0.06543	67.50560	0.0762
5400000	0.06546	67.50672	0.0762
5500000	0.06548	67.50758	0.0762
5600000	0.06547	67.50773	0.0762
5700000	0.06546	67.50526	0.0762
5800000	0.06548	67.50491	0.0762
5900000	0.06550	67.50638	0.0762
6000000	0.06552	67.50912	0.0763

Table 3.1: Convergence of the simulation for $u = 0.5$, $\mu = 1$.

N	p_B	m_V	c_V^2
100000	0.71242	225.28211	0.7073
200000	0.70856	221.70491	0.6903
300000	0.70233	219.41955	0.6978
400000	0.70205	219.53186	0.6979
500000	0.69966	217.98554	0.6955
600000	0.70234	220.52814	0.6958
700000	0.69953	218.29005	0.6974
800000	0.69836	217.09122	0.6946
$\vdots$	$\vdots$	$\vdots$	$\vdots$
5000000	0.70495	223.43205	0.7030
5100000	0.70468	223.22147	0.7028
5200000	0.70482	223.21027	0.7022
$\vdots$	$\vdots$	$\vdots$	$\vdots$
13000000	0.70641	224.86143	0.7044
13100000	0.70645	224.86849	0.7042
13200000	0.70642	224.83071	0.7041
$\vdots$	$\vdots$	$\vdots$	$\vdots$
19200000	0.70626	224.85030	0.7049
19300000	0.70628	224.85352	0.7047
19400000	0.70632	224.89686	0.7049
19500000	0.70631	224.86900	0.7046
19600000	0.70632	224.86556	0.7045
19700000	0.70637	224.89746	0.7044
19800000	0.70647	224.95485	0.7042
19900000	0.70644	224.93058	0.7041
20000000	0.70653	224.98399	0.7040

Table 3.2: Convergence of the simulation for $u = 0.9$, $\mu = 1$.

	s	$1/\mu$	m_V	$m_{V,1}$	$m_{V,2}$	c_V^2	$c_{V,1}^2$	$c_{V,2}^2$
computed	30	1	32.2989	31.0000	33.5977	0.0029	0.0010	0.0014
simulated			32.2973	30.9989	33.5954	0.0028	0.0010	0.0013
computed	60	5	71.4944	65.0000	77.9887	0.0146	0.0059	0.0065
simulated			71.4870	65.0003	77.9717	0.0145	0.0059	0.0065

Table 3.3: The limiting case $C \to \infty$, cf. Section 2.1, for $\varrho = 1.2$. In the row "simulated" the value $C = 120$ is used.

τ	C	p_B	m_V
5	8	0.32672	72.186081
2	8	0.34351	71.192965
0.5	8	0.35333	70.890139
0	8	0.35698	70.709433
5	7	0.57458	101.073861
2	7	0.59826	98.478681
0.5	7	0.60859	96.974442
0	7	0.61383	96.829807

Table 3.4: The limiting case $d_1 = 1, p_1 = 1, s = 0$ and $\tau \to 0$, cf. Section 2.2, for $\lambda = 0.1,\ \mu = 1$.

	τ	p_B
simulated	60	0.3426
	600	0.2915
	6000	0.2503
	12000	0.2480
computed	∞	0.1625

Table 3.5: Approximation of p_B for $d_1 = 1, p_1 = 1$ and $\tau \to \infty$, cf. Section 2.3, for $\lambda = 1,\ \mu = 1,\ s = 65$.

u	$1/\mu$	p_B	m_V	$m_{V,1}$	$m_{V,2}$	c_V^2	$c_{V,1}^2$	$c_{V,2}^2$
0.5	1	0.0648	67.4648	63.1220	71.8108	0.075	0.034	0.100
0.8	1	0.4242	110.6606	86.9641	134.3467	0.510	0.418	0.455
0.9	1	0.7065	224.9839	177.8439	272.1381	0.704	0.854	0.537
0.95	1	0.8620	474.0609	412.9691	535.0990	0.679	0.732	0.604
0.5	5	0.0599	75.8436	66.1944	85.4864	0.067	0.023	0.066
0.8	5	0.3688	107.5544	73.3796	141.7196	0.461	0.150	0.375
0.9	5	0.5856	167.8430	83.0473	252.6731	1.143	0.460	0.734
0.95	5	0.7332	291.9919	100.9141	483.1387	2.074	1.496	1.137

Table 3.6: Performance measures for different utilizations u and mean service times $m_B = 60 + 1/\mu \in \{61, 65\}$.

		constant retrial time			uniform distribution of R over $\left[\frac{\tau}{2}, \frac{3\tau}{2}\right]$		
u	$1/\mu$	m_V	c_V^2	p_B	m_V	c_V^2	p_B
0.69	1	82.31376	0.2676	0.23589	87.12426	0.4068	0.21389
0.85	1	148.26541	0.6510	0.56189	135.99581	0.6986	0.48602
0.90	1	224.98394	0.7041	0.70652	174.76181	0.8640	0.60848
0.80	5	107.55444	0.4613	0.36884	114.70215	0.4913	0.36723
0.91	5	181.65002	1.2964	0.61211	188.78796	1.1523	0.63217

Table 3.7: Comparison between constant retrial times $R = \tau$ and uniformly distributed retrial times R over the interval $\left[\frac{\tau}{2}, \frac{3\tau}{2}\right]$ for $\tau = m_R = 60$.

References

[AR1] Artalejo, J.R., Accessible bibliography on retrial queues. Math. Comput. Modelling 30 (1999) 223 – 233.

[AR2] Artalejo, J.R., A classified bibliography of research on retrial queues: Progress in 1990 – 1999, Top 7 (1999) 187 – 211.

[BDK] Breuer, L., Dudin, A., Klimenok, V., A retrial $BMAP/PH/N$ system. Queueing Systems 40 (2002) 433 – 459.

[CCD] Choi, B.D, Chung, Y., Dudin, A.N., The $BMAP/SM/1$ retrial queue with controllable operation modes. European J. Oper. Res. 131 (2001) 16 – 30.

[CR] Choi, B.D., Rhee, K.H., An $M/G/1$ retrial queue with a threshold in the retrial group. KYUNGPOOK Math. J. 35 (1996) 469 – 479.

[F] Falin, G., A survey of retrial queues. Queueing Systems 7 (1990) 127 – 168.

[FG] Falin, G.I., Gómez-Corral, A., On a bivariate Markov process arising in the theory of single-server retrial queues. Statistica Neerlandica 54, No. 1 (2000) 67 – 78.

[FT] Falin, G.I., Templeton, J.G.C., Retrial Queues. Champman & Hall, London, 1997.

[G] Grishechkin, S.A., Multi class batch arrival retrial queues analyzed as branching processes with immigration. Queueing Systems 11 (1992) 395 – 418.

[GH] Gross, D., Harris, C.M., Fundamentals of Queueing Theory. Third Edition, Wiley, New York, 1998.

[HLZ] He, Q.-M., Li, H., Zhao, Y.Q., Ergodicity of the $BMAP/PH/s/s+K$ retrial queue with PH-retrial times. Queueing Systems 35 (2000) 323 – 347.

[KL] Kulkarni,V.G., Liang, H.M., Retrial queues revisited. in: Frontiers in Queueing: Models and Applications in Science and Engineering, pp. 19 – 34, ed. J.H. Dshalalow, CRC Press, Boca Raton, FL, 1997.

[NR] Neuts, M.F., Rao, B.M., Numerical investigation of a multiserver retrial model. Queueing Systems 7 (1990) 169 – 189.

[S] Spahl, G., Modelling the PPP connections for a pool of modems. Informal communication, Siemens AG, Munich, 1993.

[S1] Stepanov, S.N., Numerical Methods for Calculation the Systems with Repeated Calls (in Russian). Nauka, Moscow, 1983.

[S2] Stepanov, S.N., Generalized model with repeated calls in case of extreme load. Queueing Systems 27 (1997) 131 – 151.

[YT1] Yang, T., Templeton, J.G.C., A survey on retrial queues. Queueing Systems 2 (1987) 201 – 233.

[YT2] Yang, T., Templeton, J.G.C., A survey on retrial queues (eratum). Queueing Systems 4 (1989) 94.

[W] Wolff, R.W., Stochastic Modelling and the Theory of Queues. Prentice-Hall International, Englewood Cliffs, N.Y., 1989.

[WCL] Wang, J., Cao, J., Li, Q., Reliability analysis of the retrial queue with server breakdowns and repairs. Queueing Systems 38 (2001) 363 – 381.