Finite-Source Retrial Queueing Systems with Non-Reliable Heterogenous Servers and Different Service Policies

J. Roszik, J. Sztrik
Institute of Informatics, University of Debrecen
Debrecen, Hungary

e-mail: jsztrik@inf.unideb.hu
www: http://irh.inf.unideb.hu/user/jsztrik/index.html
OUTLOOK

- The queueing model
- Applications
- Mathematical model
- Evaluation Tool MOSEL
- Case studies
- References
The queueing model
Applications

- magnetic disk memory systems
- local area networks with CSMA/CD protocols
- collision avoidance local area network modeling
Mathematical model

The system’s state at time $t$ can be described by the process

$$X(t) = (\alpha_1(t), \ldots, \alpha_c(t); N(t)),$$

where

$$N(t) = \text{the number of sources of repeated calls},$$

$$\alpha_i(t) = \begin{cases} 
1 & \text{if there is a customer under service at the server}, \\
0 & \text{if it is operational and idle}, \\
-1 & \text{if the server is failed}.
\end{cases}$$
Let us define the stationary probabilities by:

\[ P(i_1, ..., i_c, j) = \lim_{t \to \infty} P\{\alpha_1(t) = i_1, ..., \alpha_c(t) = i_c, N(t) = j\}, \]

\[ i_1, ... i_c = -1, 0, 1, \quad j = 0, ..., K^*, \]

where \( K^* = K - \sum_{i_k, i_k = 1} i_k. \)

\[ C(t) = \text{the number of busy servers}, \]

\[ A(t) = \text{the number of available servers}, \]

\[ p_{k,j} = \lim_{t \to \infty} P\{C(t) = k, N(t) = j\}. \]
Once we have obtained these limiting probabilities the **main system’s performance measures** can be derived in the following way.

- **The probability that at least one server is available**

  \[ A_S = P\{\alpha_k > -1\}, k \in \{1, \ldots, c\} = 1 - \sum_{j=0}^{K} P(-1, \ldots, -1, j). \]

- **Mean number of sources of repeated calls**

  \[ N = E[N(t)] = \sum_{k=0}^{c} \sum_{j=1}^{K} jP_{k,j} = \sum_{i_1,\ldots,i_c} \sum_{j=1}^{K^*} jP(i_1, \ldots, i_c, j). \]
• **Utilization of the $k$–th server**

\[
U_k = \sum_{i_1, \ldots, i_c, i_k=1}^{K^*} \sum_{j=0}^{K^*} P(i_1, \ldots, i_c, j), \quad k = 1, \ldots, c.
\]

• **Mean number of busy servers**

\[
C = E[C(t)] = \sum_{k=1}^{c} U_k.
\]

• **Mean number of calls staying in the orbit or in service**

\[
M = E[N(t) + C(t)] = N + C.
\]
• **Utilization of the repairman**

\[ U_R = \sum_{i_1,\ldots,i_c} \sum_{j=0}^{K^*} P(i_1, \ldots, i_c, j). \]

• **Utilization of the sources**

\[
U_{SO} = \begin{cases} 
  \frac{E[K-C(t)-N(t); A(t)>0]}{K} & \text{for blocked case,} \\
  \frac{E[K-C(t)-N(t)]}{K} & \text{for unblocked case.}
\end{cases}
\]

• **Overall utilization of the system**

\[ U_O = C + KU_{SO} + U_R. \]
• **Mean rate of generation of primary calls**

\[
\bar{\lambda} = \begin{cases} 
\lambda E[K - C(t) - N(t); A(t) > 0] & \text{for blocked case,} \\
\lambda E[K - C(t) - N(t)] & \text{for unblocked case.}
\end{cases}
\]

• **Mean waiting time**

\[
E[W] = N/\bar{\lambda}.
\]

• **Mean response time**

\[
E[T] = M/\bar{\lambda}.
\]
Evaluation Tool MOSEL

MOSEL (Modeling, Specification and Evaluation Language) developed at the University of Erlangen, Germany, is used to formulate and solve the problem.

Case studies
### Validation of results

<table>
<thead>
<tr>
<th></th>
<th>Pascal</th>
<th>random</th>
<th>FFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of servers:</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Number of sources:</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Request’s generation rate:</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Service rate:</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Retrial rate:</td>
<td>1.2</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>Server’s failure rate:</td>
<td>–</td>
<td>1e-25</td>
<td>1e-25</td>
</tr>
<tr>
<td>Server’s repair rate:</td>
<td>–</td>
<td>1e+25</td>
<td>1e+25</td>
</tr>
<tr>
<td>Mean waiting time:</td>
<td>0.1064954794</td>
<td>0.1064959317</td>
<td>0.1064959929</td>
</tr>
<tr>
<td>Mean number of busy servers:</td>
<td>1.8007480431</td>
<td>1.8007485102</td>
<td>1.8007485548</td>
</tr>
<tr>
<td>Mean number of sources of repeated calls:</td>
<td>0.1917715262</td>
<td>0.1917717923</td>
<td>0.1917718470</td>
</tr>
</tbody>
</table>
## Input parameters

<table>
<thead>
<tr>
<th></th>
<th>c</th>
<th>K</th>
<th>λ</th>
<th>$\mu_1, \ldots, \mu_c - \mu_{avg}$</th>
<th>$\nu$</th>
<th>$\delta, \gamma$</th>
<th>$\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1,5</td>
<td>4</td>
<td>20</td>
<td>x axis</td>
<td>$8,5,4,1 - 4.5$</td>
<td>4</td>
<td>0.01</td>
<td>0.2</td>
</tr>
<tr>
<td>Figure 2,6</td>
<td>4</td>
<td>20</td>
<td>4</td>
<td>$8,5,4,1 - 4.5$</td>
<td>x axis</td>
<td>0.01</td>
<td>0.2</td>
</tr>
<tr>
<td>Figure 3,4,7</td>
<td>4</td>
<td>20</td>
<td>1</td>
<td>$8,5,4,1 - 4.5$</td>
<td>4</td>
<td>x axis</td>
<td>0.2</td>
</tr>
</tbody>
</table>
$E[T]$ versus primary request generation
Mean response time versus retrial rate

$E[T]$ versus retrial rate
Mean response time versus server’s failure rate

\[ E[T] \text{ versus server’s failure rate} \]
Server utilization versus server’s failure rate

Legend
- 1. server (ordered)
- 2. server (ordered)
- 3. server (ordered)
- 4. server (ordered)
- 1. server (random)
- 2. server (random)
- 3. server (random)
- 4. server (random)

Server utilization

Server’s failure rate
Overall utilization versus primary request generation rate
Overall utilization versus retrial rate

Legend
- ordered
- random
- averaged random

Overall utilization versus retrial rate
Overall utilization versus server's failure rate

Legend
- ordered
- random
- averaged random

Overall utilization versus server’s failure rate
References


