HETEROGENEOUS FINITE-SOURCE RETRIAL QUEUES WITH SERVER SUBJECT TO BREAKDOWNS AND REPAIRS

B. Almási, J. Roszik, J. Sztrik
Institute of Mathematics and Informatics, University of Debrecen
Debrecen, Hungary

e-mail: jsztrik@math.klte.hu
www: http://it.math.klte.hu/user/jsztrik/
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

\[ X(t) = (Y(t); (\alpha C(t); \beta_1, \ldots, \beta N(t)), t \geq 0) \]

where

\( Y(t) = 0 \) if the server is up,
\( Y(t) = 1 \) if the server is down,
\( C(t) = 0 \) if the server is idle,
\( C(t) = 1 \) if the server is busy,
\( \alpha C(t) \) is the index of the request under service at time \( t \) if the server is busy.
\( N(t) \) is the number of sources of repeated calls at time \( t \),
\( \beta_j, j = 1, \ldots, N(t) \) is the indices of request staying in the orbit.
We define the stationary probabilities:

\[
P(q; 0; 0) = \lim_{t \to \infty} P(Y(t) = q; C(t) = 0; N(t) = 0),
\]

\[
P(q; j; 0) = \lim_{t \to \infty} P(Y(t) = q; \alpha_1 = j; N(t) = 0),
\]

\[
P(q; 0; i_1, \ldots, i_k) = \lim_{t \to \infty} P(Y(t) = q; C(t) = 0; \beta_1 = i_1, \ldots, \beta_k = i_k),
\]

\[
P(q; j; i_1, \ldots, i_k) = \lim_{t \to \infty} P(Y(t) = q; \alpha_1 = j; \beta_1 = i_1, \ldots, \beta_k = i_k).
\]

Once we have obtained these limiting probabilities the **main system performance measures** can be derived in the following way.
1. The server utilization with respect to source $j$

$$U_j = \sum_{k=0}^{K-1} \sum_{i_1, \ldots, i_k \neq j} P(0; j; i_1, \ldots, i_k).$$

Hence the server utilization

$$U_S = \sum_{j=1}^{K} U_j.$$ 

2. Utilization of source $i$

$$U^{(i)} = P \text{ ( source } i \text{ generates a new primary call )}.$$
3. **Utilization of the repairman**

\[ U_R = E[Y(t)] = \sum_{j=0}^{K} \sum_{k=0}^{K-1} \sum_{i_1,\ldots,i_k \neq j} P(1; j; i_1, \ldots, i_k). \]

4. **Availability of the server**

\[ A_S = 1 - U_R. \]

5. **Mean response time of source** \( i \)

\[ P_O^{(i)} = \sum_{q=0}^{1} \sum_{j=0}^{K} \sum_{k=1}^{K-1} \sum_{i\in(i_1,\ldots,i_k)} P(q; j; i_1, \ldots, i_k). \]

\[ P_S^{(i)} = \sum_{q=0}^{1} \sum_{k=0}^{K-1} \sum_{i\neq i_1,\ldots,i_k} P(q; i; i_1, \ldots, i_k). \]
\[ \gamma_i = \frac{1}{E[T_i] + E[S_i]} = \lambda_i U^{(i)} = \mu_i U_i, \quad i = 1, \ldots, K, \]

\[ E[S_i] = E[D_i] + 1/\lambda_i \geq 1/\lambda_i, \]

where \( E[D_i] \) denotes the mean delay time due to the failure of the server.

\[ U^{(i)} = \frac{1/\lambda_i}{E[T_i] + E[S_i]} = \frac{\mu_i U_i}{\lambda_i} \leq 1 - P^{(i)}, \quad i = 1, \ldots, K, \]

\[ P^{(i)} = \frac{E[T_i]}{E[T_i] + E[S_i]} = \gamma_i E[T_i] = \lambda_i U^{(i)} E[T_i], \quad i = 1, \ldots, K. \]
6. Mean waiting time of source $i$

$$E[W_i] = E[T_i] - 1/\mu_i = \frac{P^{(i)} - U_i}{\mu_i U_i}, \quad i = 1, \ldots, K.$$ 

7. Mean number of calls staying at the service facility

$$M = E[C(t) + N(t)] = \sum_{i=1}^{K} P^{(i)} = \sum_{i=1}^{K} (P_S^{(i)} + P_O^{(i)}) = \sum_{i=1}^{K} P_S^{(i)} + \sum_{i=1}^{K} P_O^{(i)}.$$ 

8. Mean rate of generation of primary calls

$$\bar{\lambda} = \sum_{i=1}^{K} \gamma_i = \sum_{i=1}^{K} \lambda_i U^{(i)} = \sum_{i=1}^{K} \mu_i U_i.$$
Evaluation Tool MOSEL

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

Case studies
### Validation of results

<table>
<thead>
<tr>
<th></th>
<th>non–rel. retrial(cont.)</th>
<th>non–rel. retrial(orbit)</th>
<th>non–rel. FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of sources:</strong></td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td><strong>Request’s generation rate:</strong></td>
<td>0.2, 0.3, 0.5</td>
<td>0.2, 0.3, 0.5</td>
<td>0.2, 0.3, 0.5</td>
</tr>
<tr>
<td><strong>Service rate:</strong></td>
<td>1, 1.2, 1.1</td>
<td>1, 1.2, 1.1</td>
<td>1, 1.2, 1.1</td>
</tr>
<tr>
<td><strong>Retrial rate:</strong></td>
<td>1e+20</td>
<td>1e+20</td>
<td>-</td>
</tr>
<tr>
<td><strong>Server’s failure rate:</strong></td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td><strong>Server’s repair rate:</strong></td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td><strong>Utilization of the server:</strong></td>
<td>0.578593008176</td>
<td>0.578593460071</td>
<td>0.578595143583</td>
</tr>
<tr>
<td><strong>Mean response time</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source 1:</td>
<td>1.61016598407</td>
<td>1.61027143737</td>
<td>1.6109393482</td>
</tr>
<tr>
<td>Source 2:</td>
<td>1.41365083148</td>
<td>1.41357129589</td>
<td>1.41287007613</td>
</tr>
<tr>
<td>Source 3:</td>
<td>1.35362123345</td>
<td>1.35362137877</td>
<td>1.35372999206</td>
</tr>
</tbody>
</table>
Mean response time depending on request generation rate

\[ E[T] \text{ versus primary request generation rate} \]
$E[T]$ versus primary request generation rate
Mean response time depending on retrial rate

$E[T]$ versus retrial rate
$E[T]$ versus retrial rate

Legend:
- homogeneous (non-continuous, intelligent)
- terminal 1 (non-continuous, intelligent)
- terminal 2 (non-continuous, intelligent)
- terminal 3 (non-continuous, intelligent)
- terminal 4 (non-continuous, intelligent)
- terminal 5 (non-continuous, intelligent)
- homogeneous (continuous, intelligent)
- terminal 1 (continuous, intelligent)
- terminal 2 (continuous, intelligent)
- terminal 3 (continuous, intelligent)
- terminal 4 (continuous, intelligent)
- terminal 5 (continuous, intelligent)
Mean response time depending on service rate

$E[T]$ versus service rate
Mean response time depending on service rate

$E[T]$ versus service rate
The graph shows the mean response time depending on the CPU failure rate in the busy state. The legend indicates different terminals with varying failure rates. The equation $E[T]$ versus CPU failure rate in busy state is also mentioned.
$E[T]$ versus CPU failure rate in busy state

Mean response time depending on CPU failure rate in busy state

Legend:
- homogeneous (non-continuous)
- terminal 1 (non-continuous)
- terminal 2 (non-continuous)
- terminal 3 (non-continuous)
- terminal 4 (non-continuous)
- terminal 5 (non-continuous)
$M$ versus retrial rate
Mean number of requests staying in the orbit or in service

Legend
- reliable CPU (heterogeneous)
- non-reliable CPU (heterogeneous, continuous)
- non-reliable CPU (heterogeneous, continuous, intelligent)
- non-reliable CPU (heterogeneous, non-continuous)
- non-reliable CPU (heterogeneous, non-continuous, intelligent)

$M$ versus retrial rate
References


