Finite-Source Retrial Queues with Applications

ICAI 2010, Eger, Hungary

27 - 30 January, 2010

Patrick Wüchner, University of Passau, Germany

János Sztrik, University of Debrecen, Hungary

Hermann de Meer, University of Passau, Germany

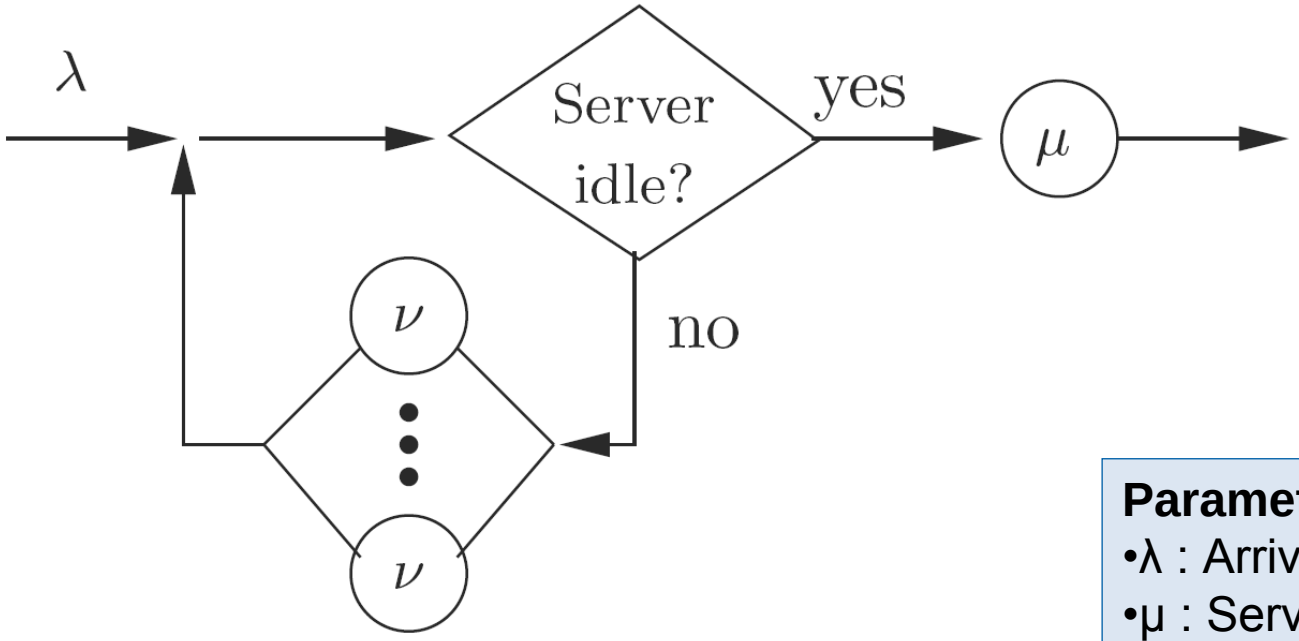
sztrik.janos@inf.unideb.hu, http://irh.inf.unideb.hu/user/jsztrik

Outline

- Basic Retrial Queueing System
- Finite-Source Model
- Applications
- Underlying Markov Chain
- Performance Measures
- Numerical Results
- MOSEL codes

Basic Retrial Queueing System

- Example: M/M/1 Retrial Queueing System



Parameters:
- λ : Arrival Rate
- μ : Service Rate
- v : Retrial Rate

Finite-source and orbital search

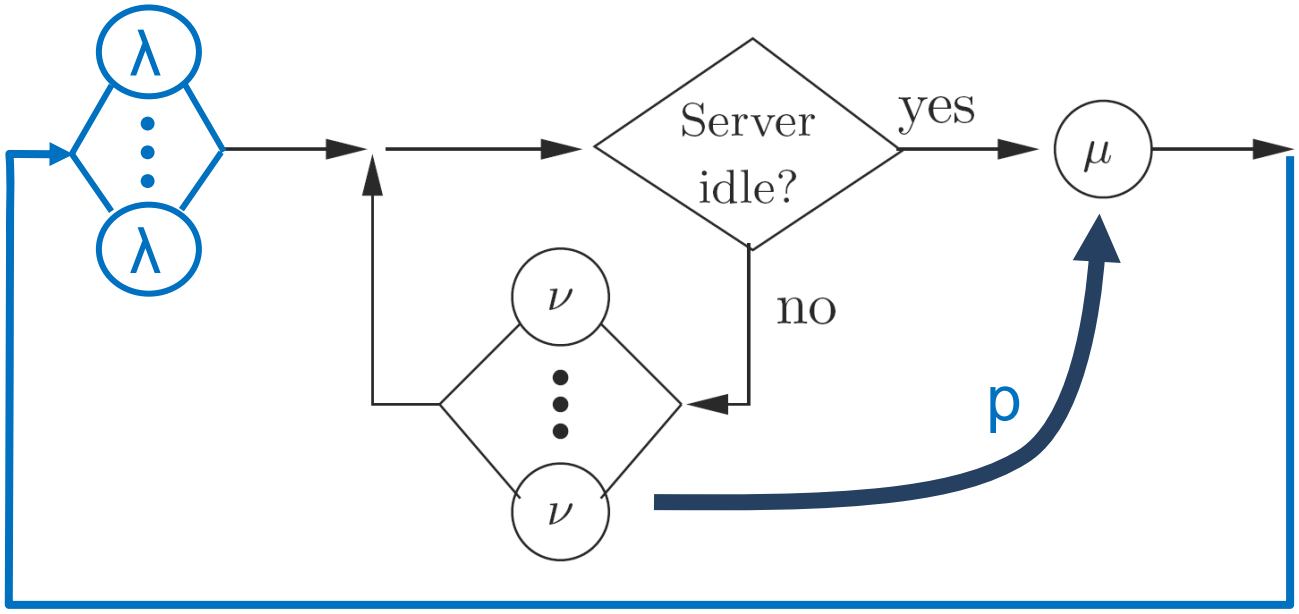
- Finite Source
  - Arrival rate depends on number of customers in the system
- Orbital Search
  - After service completion, with a certain probability, the server immediately retrieves an orbiting customer for service.

Application examples

- Computer and telecommunication systems
- Telephone networks
- Network access control
- LAN with CSMA/CD
- Call centres
- P2P file-sharing protocols
- Cellular mobile communication networks
- Magnetic disk memory systems

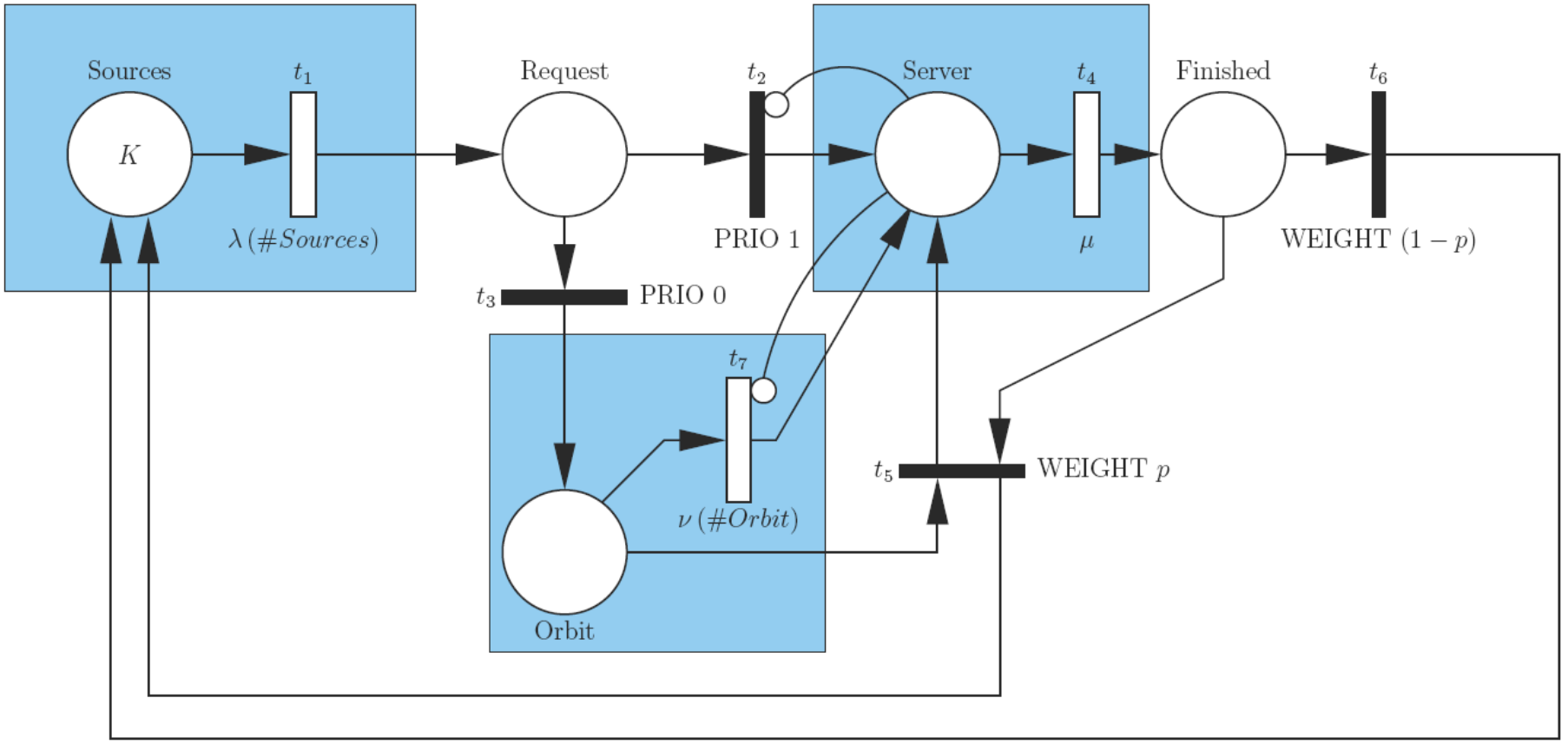
Finite-source model

- Finite Source Retrial Queue with Orbital Search



Parameters:
- K : Number of Sources
- λ : Arrival Rate
- μ : Service Rate
- ν : Retrial Rate
- p : Search Probability

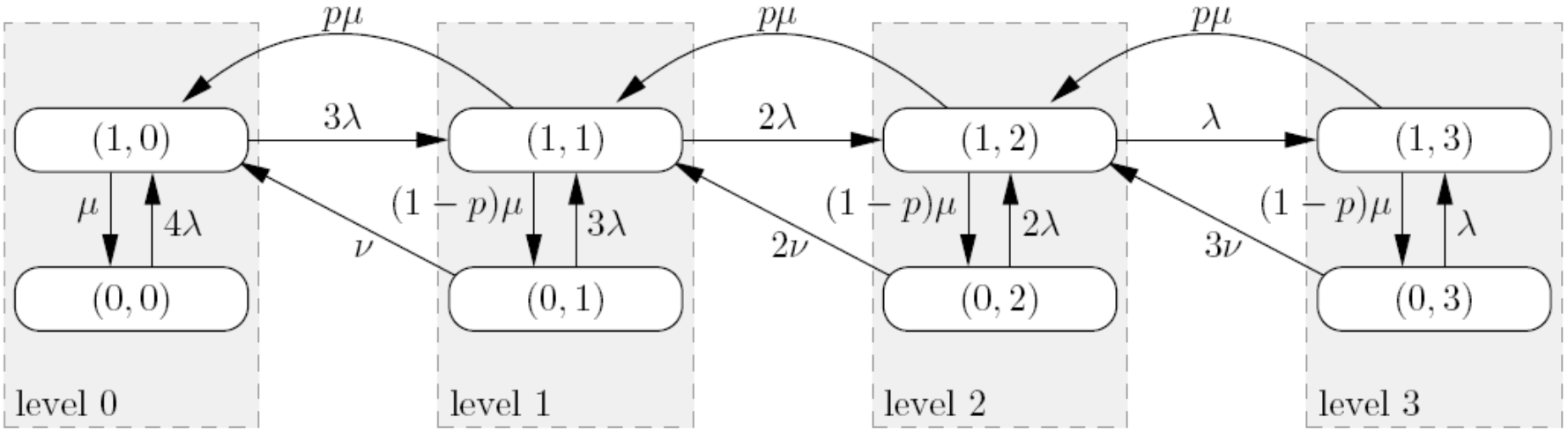
GSPN - Description



Parameters:
- K : Number of Sources
- λ : Arrival Rate
- μ : Service Rate
- v : Retrial Rate
- p : Search Probability

Underlying Markov Chain

- Two-Dimensional Continuous Markov Chain (CTMC):
  - *X(t) = (C(t), N(t)),* where *C(t)=1* if the server is busy and *0* if idle, and *N(t)* is the number of customers in the orbit at time *t*.
- Graphical Representation of CTMC for K=4:



Parameters:
- μ : Service Rate
- K=4 Sources
- v : Retrial Rate
- λ : Arrival Rate
- p : Search Probability

Performance Measures (1)

- Stationary Probabilities

$$P(r,j) = \lim_{t\to\infty} P(C(t) = r, N(t) = j), \quad r = 0,1, \quad j = 0,...,K-1.$$

- Server Utilization

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

- Mean Number of Customers in the Orbit

$$N = \sum_{r=0}^{1} \sum_{j=0}^{K-1} jP(r,j)$$

- Mean Number of Customers in the System (Service and Orbit)

$$M = \sum_{r=0}^{1} \sum_{j=0}^{K-1} (r+j)P(r,j)$$

Performance Measures (2)

- Mean Generation Rate of Primary Calls

$$\overline{\lambda} = \lambda(K - M)$$

- Mean Response Time

$$E[T] = M/\overline{\lambda}$$

- Mean Waiting Time

$$E[W] = N/\overline{\lambda}$$

- Blocking Probability of Primary Calls

$$B = \frac{\lambda E[K - C(t) - N(t), C(t) = 1]}{\overline{\lambda}}$$

Numerical Results (1)

- Obtained using MOSEL-2 with SPNP
- Parameters set:

Parameter	Symbol	Value / Range
Number of sources	K	3
Service rate	μ	1
Generation rate	λ	0.0001 … 1
Retrial rate	ν	0.001 … 0.5
Search probability	p	$1 \cdot 10^{-8}$

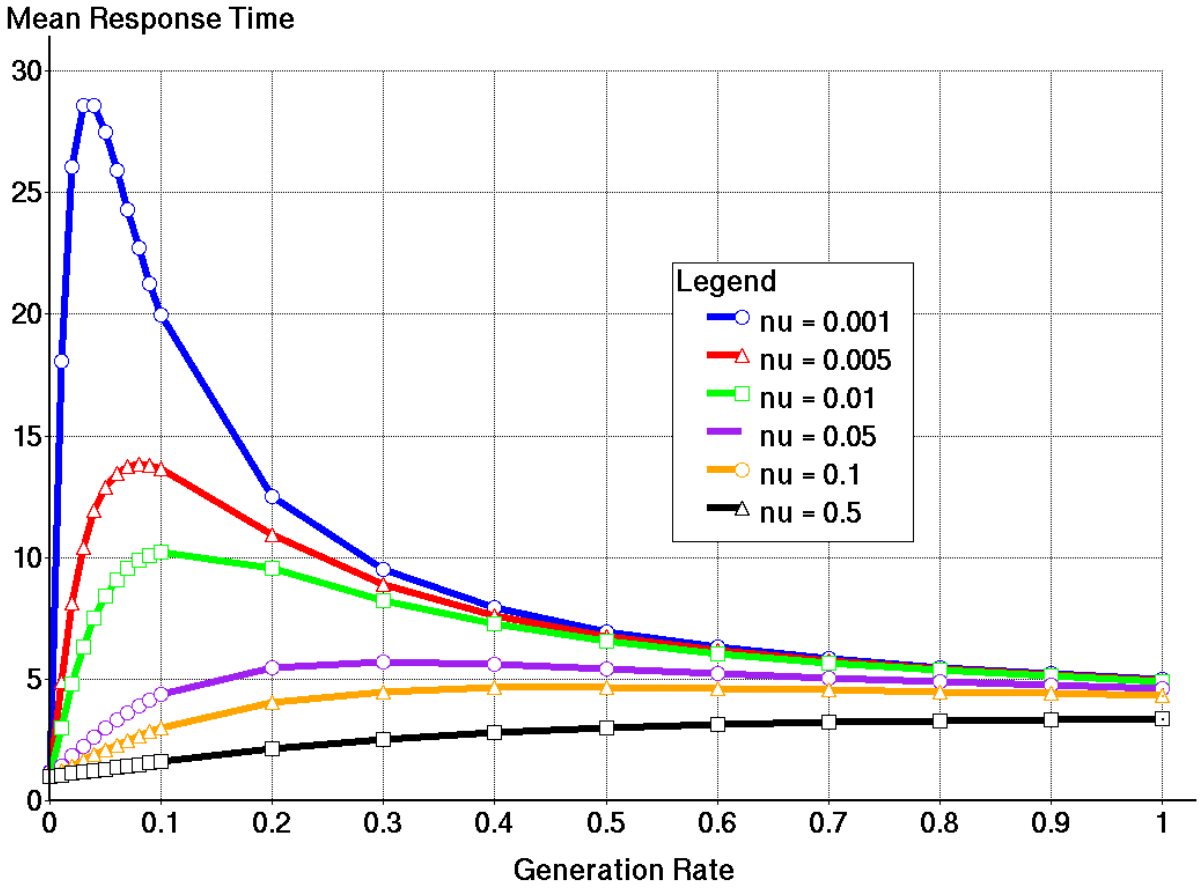
Numerical Results (2)

- Obtained using MOSEL-2 with SPNP
- Parameters set:

Parameter	Symbol	Value / Range
Number of sources	K	3
Service rate	μ	1
Generation rate	λ	0.0001 … 1
Retrial rate	ν	0.005
Search probability	p	$1 \cdot 10^{-8}$ … $(1-1 \cdot 10^{-8})$

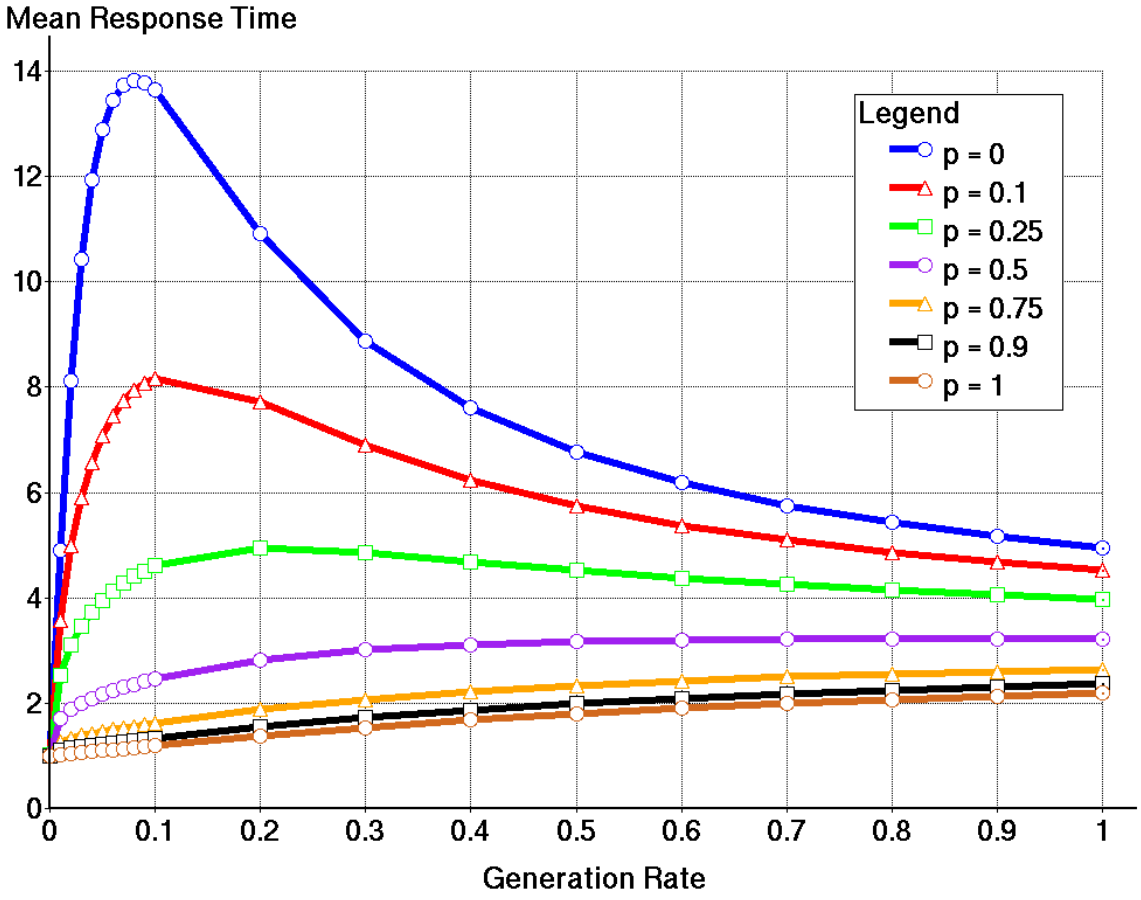
Numerical Results (3)

- Mean response time versus arrival rate (influence of *v*)

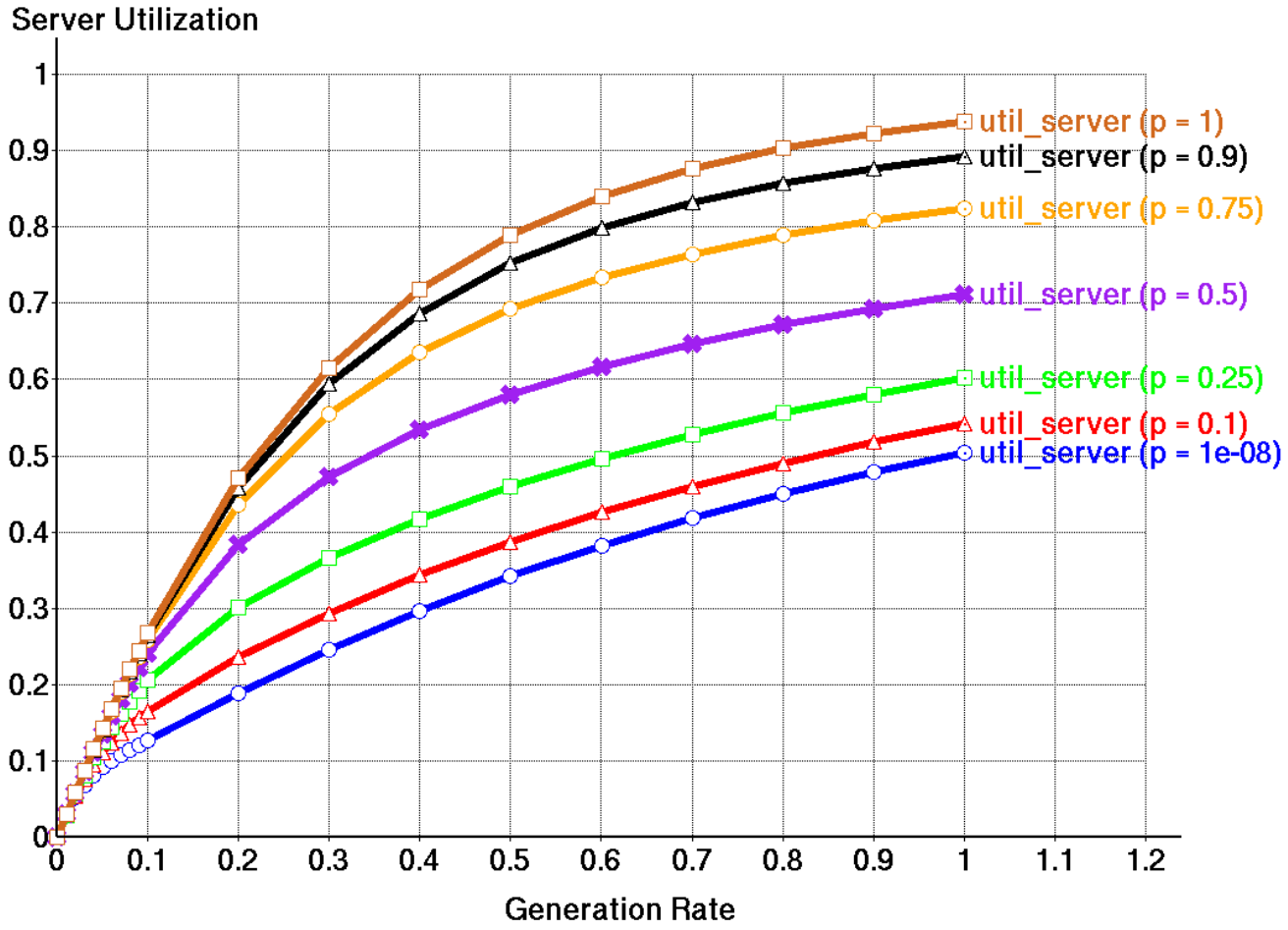


Numerical Results (4)

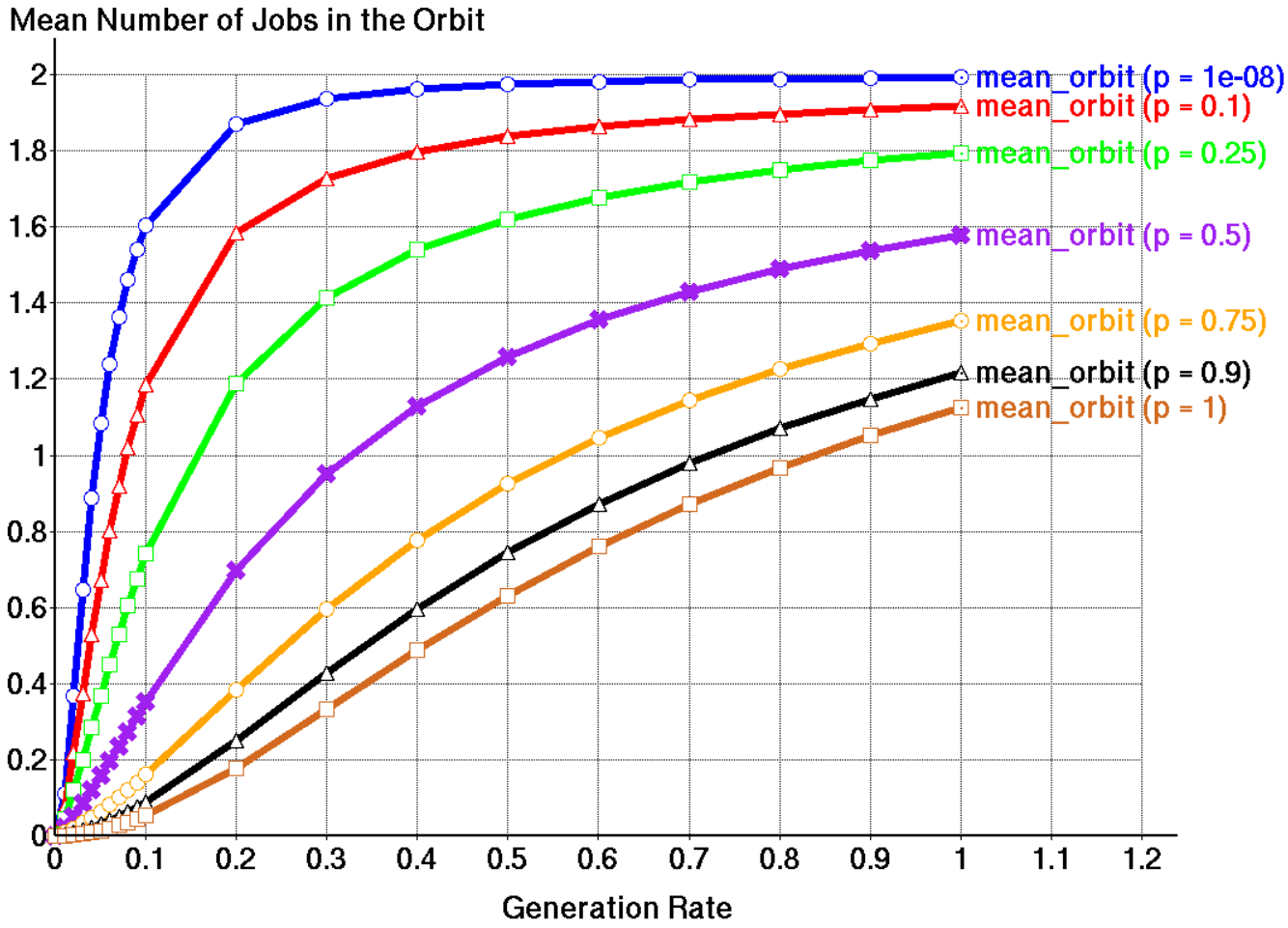
- Mean response time versus arrival rate (influence of *p*)



Additional Numerical Results (1)



Additional Numerical Results (2)

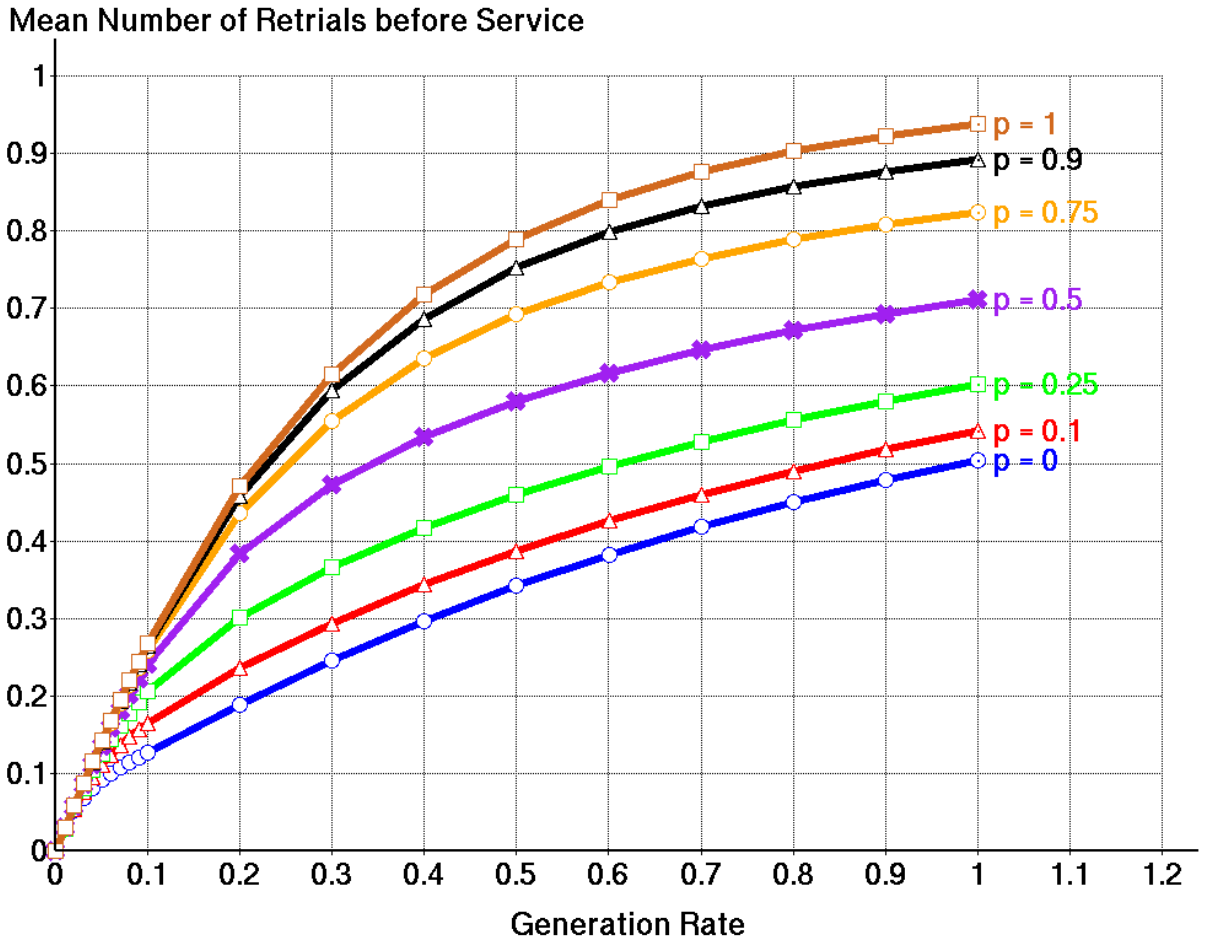


Additional Numerical Results (4)

- Mean number of retrials before service

  - $E[T] = 1/\mu + E[T_{Orbit}]$
  - $E[T_{Orbit}] = E[T] - 1/\mu$
  - $\#Retrials = E[T_{Orbit}] \; / \; (1/\nu) = E[T_{Orbit}] \cdot \nu$

Additional Numerical Results (5)



Generator Matrix

$$\mathbf{Q} = \left(\begin{array}{cc|cc|cc|cc} -4\lambda & 4\lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu & -\mu-3\lambda & 3\lambda & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & p\mu & -\mu-2\lambda & (1-p)\mu & 2\lambda & 0 & 0 & 0 \\ 0 & \nu & 3\lambda & -\nu-3\lambda & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & p\mu & 0 & -\mu-\lambda & (1-p)\mu & \lambda & 0 \\ 0 & 0 & 2\nu & 0 & 2\lambda & -2\nu-2\lambda & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & p\mu & 0 & -\mu & (1-p)\mu \\ 0 & 0 & 0 & 0 & 3\nu & 0 & \lambda & -3\nu-\lambda \end{array}\right)$$

MOSEL-2 Model

```
1  /*** CONSTANTS AND PARAMETERS ********************************************************/
2  CONST      K       := 3;                                          // population size
3  CONST      mu      := 1;                                          // service rate
4  PARAMETER lambda := 0.0001, 0.001, 0.01 .. 0.1 STEP 0.01;         // request gen. rate
5  PARAMETER nu       := 0.001, 0.0025, 0.005;                       // retrial rate
6  PARAMETER p        := 0.00000001, 0.00001, 0.01;                  // search probability
7
8  /*** NODES ***************************************************************************/
9  NODE Sources[K]   := K;                                           // sources
10 NODE Request[1]   := 0;                                           // generated call
11 NODE Server[1]    := 0;                                           // the server
12 NODE Orbit[K]     := 0;                                           // retrying customers
13 NODE Finished[1]  := 0;                                           // serviced customer
14
15 /*** RULES ***************************************************************************/
16 FROM Sources    TO Request   RATE Sources*lambda;                 // primary requests
17 FROM Request    TO Server    PRIO 1;                              // to server if idle
18 FROM Request    TO Orbit     PRIO 0;                              // to orbit if busy
19 FROM Orbit      TO Server    RATE Orbit*nu;                       // retrials
20 FROM Server     TO Finished  RATE mu;                             // service
21 FROM Finished   TO Sources   WEIGHT 1-p;                          // without polling
22 FROM Finished, Orbit TO Server, Sources WEIGHT p;                 // with polling
23
24 /*** RESULTS *************************************************************************/
25 PRINT util_server  := UTIL(Server);                               // server utilization
26 PRINT mean_lambda  := MEAN(Sources)*lambda;                       // mean arrival rate
27 PRINT mean_T       := (K-MEAN(Sources))/mean_lambda;              // mean response time
28 PRINT mean_orbit   := MEAN(Orbit);                               // mean cust. in orbit
```

Bibliography

- Almási B. – Roszik J. – Sztrik J. Homogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Mathematical and Computer Modelling* 42(2005), 673-682
- Begain K. – Bolch G. – Herold H. *Practical Performance Modeling*, Kluwer Academic Publisher, 2001
- Artalejo J.R. – Comez-Corral A. *Retrial Queueing Systems*, Springer, 2008
- Falin G.I. – Templeton J.G.C. *Retrial Queues*, Chapman and Hall, London, 1997
- Hayes J.F. – Babu T.V.J. *Modeling and Analysis of Telecommunication Networks*, Wiley-Interscience, 2004


Thank You
for Your
Attention