A Queuing Model with Markovian Arrival and LCFS Priority System and General Time for Service

Yogesh Kumar Bhardwaj*, K. M. Sharma
BSA college of Engineering & Technology, Mathura (U.P.), INDIA

ABSTRACT: We consider a single server queuing system with repeated attempts in which customers arrive according a Markov arrival process (MAP) and with a LCFS PR discipline. The service times are independent and have a common general distribution. After service completion time the server initiates his search time with an arbitrary distribution function. We consider two cases where the maximum number of repeated customers waiting in the orbit to seek service again is limited by r (r<∞) or can be unlimited (r=∞). We derive the steady state probabilities of the embedded Markov chain at service completion time of the process and also steady state probabilities of underlying Markov’s linear process.

Keywords: Queuing theory, Markov arrival process (MAP), LCFS PR discipline, IP multimedia subsystem, Markov Chain.

INTRODUCTION: Recently, internet services based on the session notion have grown in number and popularity. Examples of such services are multimedia conferences, IP telephony, instant messaging, and similar multimedia applications. Deployment scenarios include session control services within the IP multimedia subsystem (IMS), in the 3rd generation mobile networks. In the IMS, the call session control function (CSCF) servers perform session management, based on the session initiation Protocol (SIP). Session control protocols such as SIP are transactional protocols. In general, a transaction consists of a single request, any intermediate provisional response, and a final response to that request. Fault - tolerance in session control systems is achieved by introducing redundancy, e.g., through reliable server pooling or clustering. Namely, session control servers are multiplied in server sets. Session control is a time - critical applications.

Performance of session control is quantified by transaction control time. Transaction control time is defined as the mean time between the moment of request sending and the moment of final response receipt at the transaction initiator (including possible multiple fail - overs to different servers). One important challenge in such replicated session control system is how to enhance performance, i.e., how to reduce transaction control time. The SSPs are crucial in reducing transaction control line. The main problem addressed in this paper is maximizing the probability of successful transaction with the current transmission, thereby minimizing the average number of attempted servers until success. The ultimate achievement is reduced transaction control time. The proposed method is most useful in cases of frequent intersection between the client and the same server set (e.g. in messaging session utilizing SIP servers) as the status for a given server set stored in the client is in that case always maintained up - to - date. Note however that depending on the SIP architecture and the deployed fault tolerance mechanisms, it is very well possible that the role of the client is taken by a Proxy server e.g. the P-CSCF in 3GPPIMS, that as a consequence may interact very frequently with the deployed replicated server set ; in case of 3GPPIMS the latter could correspond to the S-CSCF . Second, the scheme requires adaptations to the clients, but large parts of those can be implemented in standardized protocol layers.

Neuts and Ramalhoto were the first introduced the queuing model with customer - searching server. In (Neuts et al 1984) they considered the system M/G/1 with infinite orbit. The system M/1/G/1/r with a finite buffer and priority search for customers is analyzed in [Neuts, 1981], where recurrent formulas for stationary state probabilities of the system are derived. In [Latouch and Ramaswami (1993)]; Neuts and Ramalhoto (1984)] the systems M/HG/1/r/s and M/HG/1/r/s with one dimensional or pluri - dimensional poisson flow , finite buffer and orbit from which retrial customers make attempts to rejoin to the queue in buffer. In [Atencia, 2001] the system M/G/1 with LCFSPR discipline of service and general searching times is studied. In this paper we extended the obtained results in [Atencia, 2001] to the case where the flow of customers in MAP. It is worth pointing out that the method is not limited to only session statefull servers, i.e. stateless servers multiplied in a server pool for fault-tolerance reasons (higher availability /reliability) also benefit from this method. This means that the servers in the same server pool do not necessarily need to maintain and replicate
session states. Thus, the method is applicable to both stateless sets and session statefull server sets. Round Robin (RR) is a cyclic policy, where servers are selected sequentially in cycle [Mauve (2000)] weighted Round Robin is a simple extension of round robin. It assigns a certain weight to each server. The weight indicates the server processing capacity. This SSP may also be dynamic if it can evaluate individual server’s, capacities and their loads occasionally [Campbell, et al (2002)]. The unawareness of dynamic system states leads to low complexity, however, at the expense of potentially degrading performance and service dependability. Dynamic (adaptive) SSPs make decisions based on changes in the system state the dynamic estimation of the best server. Least used SSP [Henrik (2003)]. In this SSP, each server load is monitored by a central monitoring entity or by the client itself. Based on monitoring the load of servers, each server is assigned the so-called policy value, which is proportional to the server’s load. According to the least used SSP, the server with the lowest policy value is selected as the receiver of the current message. It’s important to note that this SSP implies that the same server is always selected until the policy values of the servers are updated and changed. Thus, this server may no longer have the lowest policy value in the server set an the policy may overtime converge towards round robin. Every update of the servers’ policy values brings the SSP back to least used with degradation. The effectiveness of a dynamic SSP critically depends on the matrix that is used to evaluate the best server. The research on SSPs has been mainly focused on the replicated web server systems. In such systems, the typical metrics are based on server proximity including geographic distance, number of hops to each server, round trip time (RTT) and HTTP response times [Bozinovski et al. (2003), [Camarillo et al. (2003)], [Obrazcka and Silvia, (2000)]. The main objective of SSPs in web systems is providing small service latency and high throughput (i.e. successfully transmitted data volumes in the short time periods). This metrics are particularly important as the web service is based on downloading files from web servers over the internet.

Downloads can take substantial amounts of time if servers responsiveness and link band width are not optimized. While SSPs in web systems aim to provide high throughput and small service latency, session control protocols such as SIP deal with message being rather small is size 500 bytes on average [Bozinovski et al.(2004); Rosenberg (2002)]. Thus throughput as measured in data volumes per time unit is not an as significant metric as in the web systems. However, service latency remains an important performance parameter. To best of authors; knowledge, SSPs have not been extensively investigated in the context of replicated session control systems.


The paper is organized as follows. The existing server selection policies are discussed in section II. In section III a formal description of the proposed MA server selection algorithm as well as implementation details are presented. Furthermore, alternatives of proposed MA SSP are outlined. The protocol extension to integrate MA into RSet pool architecture is proposed in section IV.

The system model assumptions used for derivation of analytic expressions and simulation programs are described in section V. The definitions of the evaluation metrics, the actual derivation of the analytic model, and the description of the event-driven simulation programs are presented in section VI. Evaluation of the novel MA SSP and discussion of numerical results obtain via analytic expressions and simulation is performed in the section VII. Concluding remarks and direction selection policies have been extensively studied in the literature. Some currently existing static and dynamic algorithms are outlined in this section existing static server selection policies use predefined schemes for selecting servers.

2. Mathematical Formulation: We consider a single-server queueing system without buffer. Customers’ flow entering the system forms a Markov Arrival process (MAP), defined by matrices A and N of the size 1. The entry Ai,j, i ≠ j of matrix A is the transition intensity of the generation process from the phase i to the phase j with no new arrivals, while the entry Ni,j of matrix N is the transition intensity from phase i to phase j with new arrival.

Additionally, we introduce the matrices A*=A+N and suppose that the matrix A* is conservative matrix of the transition rates of the controlled Markov process.
\{\xi(t), t \geq 0\} in the space of the generative process \{1, 2, \ldots, \} and \(N\) is not null matrix.

The service time of consumers are independent random variables with a common arbitrary distribution function (DF) \(B_0(x)\). When the server become free, that is when service is completed, the server start his search time even of the orbit is empty, with an arbitrary distribution function \(B_0(x)\). An arriving customer finds the server idle obtains service immediately or expects the costumer in the service; otherwise, he begins his service. The expelled costumer join a group of unsatisfied costumers, that we called orbit, in order to get service later. At a search completion moment the server starts new service if there are costumers on the orbit or makes new search if the orbit is empty.

Assume that \(B_0(0) = 0, S=0,1\)

And \(\int_0^\infty t dB_s(t) = \frac{1}{\mu_s} \infty\)

A costumer that arrives when the server is busy and the orbit is occupied fully removes the costumer standing in service from the server and starts his own service.

**The Case of Finite Orbit:**

In this section we consider the case where the maximum number of customers in the orbit are limited by number \(r, (r\rightarrow \infty)\).

**The Embedded Markov Chain:**

Firstly we introduce auxiliary matrix \(B_s, K_s, s = 0, 1\) that will be needed for further consideration

\[ B_s = \int_0^\infty e^{A x} dB_s(x) \]

\[ K_s = \int_0^\infty e^{A x} N(I - B_s(x))dx \]

\[ = -(I - B_s)A^{-1}N \]

The matrix entry \((K_s)_{i,j}\) is the probability of successful completion of searching (S=0) or serving (S=1) process given that the generation passed to the phrase \(j\) at the completion moment and started in the phase \(i\) at the beginning.

The matrix entry \((K_s)_{i,j}\) is the probability of interruption of searching (S=0) or serving (S=1) process by a new arrival costumer given that the generation passed to phase \(j\) at the interruption moment and started in the phrase \(i\) at the beginning.

Now we consider the embedded Markov chain \(\gamma_{K}\) \(K \geq 0\) induced by the moments immediately after changes of server states (interruption or completion of searching or serving process). The Markov chain \(\gamma_{K}\) \(K \geq 0\) has the following set of states

\[ y = \{ (i, s, n), i = 1, \ldots, r, s = 0, 1, n = 0, \ldots, r \} \]

The state \((i, s, n)\) means that at time \(t\) there are \(n\) costumers in the orbit, the generation process is in the phase \(i\), the server operates on the searching \((s = 0)\) or serving \((s = 1)\) regime. We denote by \(\pi_{s,n}\) the stationary probabilities of state \((i, s, n)\) and introduce their row derivative vectors.

\[ \Pi = (\Pi_{1,0}, \ldots, \Pi_{1,n}, \Pi_{2,0}, \ldots, \Pi_{r,0}, \ldots, \Pi_{r,n}) \]

Let

\[ Q_{(i,s,n)(i',s',n')} = \text{Prob}(\gamma_{K} = (i', s', n') | \gamma_{K,1} = (i, s, n) \]

We define the probability transition matrix \(Q = (Q_{s,n})_{s,n} = 0, r\) for the embedded Markov chain \(\gamma_{K} \geq 0\):

\[ Q = \begin{bmatrix} Q_{0,0} & Q_{0,1} & 0 & \cdots & 0 & 0 \\ Q_{1,0} & Q_{1,1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & Q_{2,1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Q_{r,r-1} & Q_{r,r} \end{bmatrix} \]

Now we denote by

\[ \Pi = (\Pi_{0,1}, \ldots, \Pi_{r,r}) \]

The vector of stationary probability of Markov chain. The system of equilibrium equation (SEE) can be written in the form \(\Pi Q = \Pi\) with the normalization condition \(\Pi 1 = 1\)

For solving SEE we introduce the \(n\)-system, which is obtained from the original system by observing its behavior on the states when the number of customers in the orbit don’t exceed \(n\). Consideration of relation between the \(n\)-systems and the original system help us to solve SEE by the following way.

First we transform matrix \(Q\) into matrix \(Q'\). In each step \(m\), where \(m\) runs from \(r\) to \(1\), we transform the principle block-matrix elements of the matrix \(Q\) are zeros. Therefore the matrix \(Q\) has the form

\[ Q = \begin{bmatrix} Q_{0,0} & Q_{0,1} & 0 & \cdots & 0 & 0 \\ Q_{1,0} & Q_{1,1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & Q_{2,1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Q_{r,r-1} & Q_{r,r} \end{bmatrix} \]

The vector of stationary probability of Markov chain. The system of equilibrium equation (SEE) can be written in the form \(\Pi Q = \Pi\) with the normalization condition \(\Pi 1 = 1\)

For solving SEE we introduce the \(n\)-system, which is obtained from the original system by observing its behavior on the states when the number of customers in the orbit don’t exceed \(n\). Consideration of relation between the \(n\)-systems and the original system help us to solve SEE by the following way.

First we transform matrix \(Q\) into matrix \(Q'\). In each step \(m\), where \(m\) runs from \(r\) to \(1\), we transform the principle minor of size \(m\) by the following changes:

Matrix \(Q_{m,m}\) by

\[ Q_{m,m} = (I - Q_{m,m})^{-1} \]

Matrix \(Q_{m,m-1}\) by

\[ Q_{m,m-1} = Q_{m,m} Q_{m,m-1} \]

Matrices \(Q_{m,m}, K = m-2, m-1, \) by

\[ Q_{K,m,m} = Q_{K,m-1} + Q_{K,m} Q_{m,m-1} \]
All the other matrices $Q_{n,n}$ are not changed. For compact exposition we will not introduce new notations we keep the old name for the obtained after each step matrix $Q'$ as $Q$. The proof of invertibility of the matrix $I-Q_{n,n}$ is straightforward.

The process continue up to the minor $Q_{0,0}$. After this, we solve the SEE to within a constant factor: $\pi_0 = \pi_0 Q_{0,0}$.

Then, we use the recurrent formulas to establish the vectors $\pi_n$, $n = 1, 2, n = 1, 2, 3, \ldots$ to within a constant factor:

$$\pi_n = \pi_{n-1} Q_{n-1,n} Q_{0,n}.$$  

Finally we define the values of the constant using the normalization condition.

Linear Markov Process and its steady state distribution

The behavior of the investigated queueing system can be described by a linear Markov process

$$\{\gamma(t), t \geq 0\} \text{ for the linear Markov process}$$

We define by $P_i$ - stationary probability of state $(i,0)$; by $P_{i,0}(x)$ - stationary probability density of state $(i,0,0)$; and by $P_{i,s,n}$ - stationary probability of state $(i,s,n,x)$ with no regards on the elapsed time $x$. We introduce also row vectors of stationary probability $\vec{P}_{i,s}(x) = (P_{i,s,0}(x), \ldots, P_{i,s,n}(x))$.

Knowing the stationary probabilities of states of embedded Markov chain and using the methods of the renewal theory we can define the stationary distribution for the linear Markov process $\{\gamma(t), t \geq 0\}$. For this purpose we consider the mean time $\bar{\xi}$ of the interval between two adjacent changed moments of the embedded Markov chain. In the stationary system mode $\bar{\xi}$ can be defined by the formulas

$$\bar{\xi} = \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} \pi_{r,s} \frac{1}{r} \frac{1}{s} + \pi_s.$$  

where $\pi_s - \int_0^\infty dB_s(t)$ – the mean time of successful searching $(s=0)$ or serving $(s=1)$ process.

Therefore the stationary probability density $\vec{P}_{i,s}(x)$, $s = 0,1$, can be written in the term of $\pi_{r,s}$ - stationary probabilities of embedded Markov chain:

$$\vec{P}_{i,s}(x) = \frac{1}{\bar{\xi} e^{\lambda x}} e^{\lambda x} (1, B_i(x))$$

Finally we get the stationary probabilities of system states without regards on the elapsed time

$$\vec{P}_{i,s}(x) = \frac{1}{\bar{\xi} e^{\lambda x}} e^{\lambda x} (1, B_i(x)) dx = \frac{1}{\bar{\xi}} \pi_{r,s} e^{\lambda x} (1, B_i(x))$$

3. The case of Infinite orbit

In this section we consider the same queueing system as in the first section, but now the length of orbit is infinite. In this case the probability transition matrix $Q$ of embedded Markov chain $\{\gamma(t), t \geq 0\}$, has the following form

$$Q = \begin{bmatrix}
Q_{1,0} & Q_{1,1} & 0 & \cdots \\
Q_{2,0} & Q_{2,1} & Q_{2,2} & \cdots \\
0 & Q_{2,1} & Q_{2,2} & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
\end{bmatrix}$$

we denote by $n$ level of subset of the states of embedded Markov chain, where the number of orbiting customers in the orbit in $n$, and consider the matrix $R_{n,n+1}$, the entries of which are the probabilities of the first to $n-1$ level from a state of $n$ level; at same time the phases changed in correspondence with indexes. Having $R_{n,n+1} = Q_{n,n+1} + q_{n,n} R_{n,n+1} + Q_{n,n+1} R_{n,n+1} + q_{n,n+1}$ and rewrite the equation for $R$ in the term $Q = A+B+CR$, where matrix $D = A+B+C$-transition matrix of a Markov Process. Suppose that the matrix $D$ is irreducible.

The solution of this well-known equation can be found in different works. For example [Bocharov, et al. (2000); Neuts(1981)]. Suppose that the matrix $R$ is known. We consider the $r$- system, which is obtained from the original system by observing its behavior on the states when the number of customers on the orbit does not exceed $r$. The $r$-system differ from the original system by the limited length $r$ of orbit. The transition matrix $Q$ for embedded Markov chain $\{\gamma(t), t \geq 0\}$, differ from the transition matrix $Q$ of embedded Markov chain, describe in the first section, in matrix elements $Q_{r,n}$ only

$Q_{r,n} = Q_{r,n} + Q_{r,n+1}$

Therefore we have similar $r$-system as in the section 2. The stationary probabilities of this system differ from the stationary probabilities of the original system in a constant only. Further analysis of the system does not differ from the analysis in the section 2 and it is not repeated.

4. Numerical results

The sceneries considered can be classified into the following categories(1) Analytic model validation results based on the analytic expressions in section 3 and result based on simulation model in section 3. Within the section, all analytic model is validated through a detail quantitative comparison between results based on
simulation model 3. The input parameters are presented in table 1.

Fig 1 and 2 present the transaction dependability and the transaction control time as function of the inter-transaction time, respectively. The inter-transaction time is a determinative variable in this scenario. The numerical results are obtained via simulation and through the analytic expressions drive from MA and with deterministic inter-transaction time. The following important conclusions are drawn.

1. In fig.1, it is interesting to note that the transaction dependability matrices achieve the maximum inter-transaction time approaches zero. In general this is valid for such a model because the effective inter-request time is always lower-bounded to NT. When the entire server is in the OFF interval. On the other hand, there is no lower bound on the effective inter-transaction time when there is at least one server is ON interval. Thus, shorter effective inter-transaction time implies that the percentage of transaction that arrives in an ON interval is growing as opposed to the percentage of transaction that arrives in OFF interval.

2. The transaction dependability metrics converge towards the limit probability of 0.9975, which is obtained as $1 - (1 - P)^2$ when the inter-transaction infinitely grows (fig.2); where $P$ is the a priori availability of each particular server (i.e., 0.95 in this particular scenario). For extremely large inter-transaction times, the correlation between two subsequent transactions in negligible.

3. The curves of the transaction dependability matrices for either SSP are very close to each other (fig.2). The is intrusively expended, as the sequence order in which the servers are accessed only has a small impact on the probability of transaction success for the considered scenarios that the time-scales of the failure/repair model are much larger than the time scales of an individual transaction. The MA SSP is slightly more efficient as it attempts to maximize the probability of successful transaction with every transmission.

4. The INVITE transaction time is generally longer than the IM/BYE transaction time due to the larger number of messages to be exchanged with the INVITE transaction (fig.1).

5. A novel dynamic server selection policy SSP referred to as maximum availability (MA) SSP is proposed along with a simple RSerPool protocol extension. MA is applicable in a board range of IP-based systems and services, even though in this paper it was only considered in session control systems. The proposed MA SSP is a dynamic and adaptive algorithm that aims at maximizing the probability of successful transaction with the current transmission. Decisions are made using a simple status vector maintained at a client. MA has a low implementation complexity; a client should only keep a status vector with as many element as servers in the server set.

An analytic model was developed and expression of the selected evaluation matrices was derived. Also, based on the system model assumptions, event-driven simulation programs were developed. By comparing the results obtain from the expressions and simulations, the analytic model was validated.

Numerical results show that the MA significantly out performs RR. Moreover, for the considered system model, MA without a heat beat Mechanism support gives the best service dependability and performance in non-homogenous server sets.

In order to integrate MA into RSer Pool, a simple RSer Pool extension was proposed, where a name resolution response is added an extra field containing the status vector. Nevertheless, MA is Not restricted to RSer Pool, and may be used with any other fault-tolerant platform such a clusters. In the future work, dynamic SSPs based on other metrics will be devised. For instance, in order to further enhance performance, application response time (or the round trip time) from each server can be measured to assess each server’s processing power. In this paper uptime and downtime were assumed exponentially distributed. It would be interesting to also consider other failure/repair models as well as other traffic models. The performance of MA may differ for distributions other than exponential. The current analytic model was derived only for MA. In the future studies, we aim at extending it to other SSPs with or without HB support. Comparison of SSP to a standard setup with more realistic parameter setting (e.g. SCTP heart-beat mechanism with appropriate parameters) for telephony signaling will be performed in the future work. Finally, assuming synchronized time (between PU and ENRP servers) may be an unreasonable constraint. The impact of varying clock skew on the SSP, and the possible mechanisms that could alleviate the clock skew breakage will be investigated and evaluated in the future studies.

**TABLE 1: INPUT PARAMETERS**

<table>
<thead>
<tr>
<th>N</th>
<th>$t_n^{(SSP)}$</th>
<th>$t_{sip}$</th>
<th>$t_{pu}$</th>
<th>$t_{pe}$</th>
<th>$1/\lambda_{on}$</th>
<th>$1/\lambda_{off}$</th>
<th>$T_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.01s</td>
<td>50ms</td>
<td>4750s</td>
<td>250s</td>
<td>1s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

107 | Yogesh Kumar Bhardwaj, K. M. Sharma
Figure 1: INVITE transaction time versus normalized session arrival rate

Figure 2: INVITE transaction dependability versus normalized session arrival rate

6. References