

Quantitative Analysis of Single-Level Single-Mediator Multi-Agent Systems

Moon Ho Lee¹, Aliaksandr Birukou³, Alexander Dudin^{1,2}, Valentina Klimenok², and Chang-hui Choe¹

¹ Institute of Information and Communication
Chonbuk National University
Chonju, 561-765, Korea
moonho@chonbuk.ac.kr

² Department of Applied Mathematics and Computer Science
Belarusian State University
Minsk, 220030, Belarus
dudin@bsu.by, klimenok@bsu.by

³ Department of Information and Communication Technology
University of Trento
Povo(Trento), 38050, Italy
birukou@dit.unitn.it

Abstract. Queueing Theory deals with problems where some restricted resource should be shared between competitive flow of requests. In this paper we use Queueing Theory methods to perform a quantitative analysis of a single-level single-mediator multi-agent system. In the system, several agents, coordinated by the mediator process user queries. We adopt matrix analytic methods to compute performance characteristics in terms of a queueing network of tree-like topology with cooperation of the servers. Results can be used for the logical and technical design and optimal resources sharing in multi-agent systems.

1 Introduction

Organizational structures is a popular research direction in the field of Multi-Agent Systems (*MASs*), see e.g. [6, 8]. However, there are only few papers dealing with quantitative analysis of such systems [1, 5]. Queueing theory (*QT*) investigates situations when some restricted resource should be efficiently shared between competitive flow of requests in an optimal way. So, definitely, it should be useful in quantitative investigation and comparison of different organizational structures of *MASs*. Possibility of *QT* applications for *MAS* was discussed, e.g., in [1],[2]. For instance, in [2] operation of *MAS* is described in terms of queueing networks. In [1], the *M/M/1* queueing system is used for utility prediction for a range of possible *MASs*.

In particular, *QT* is an appropriate tool for qualitative analysis of single-level single-mediator *MASs* [1]. In such class of systems the agent called mediator dis-

tributes queries⁴ among several agents. In this paper we perform a quantitative analysis of single-level single-mediator *MASs* using *QT*. The results allow us to calculate such performance characteristics as probability of query being rejected, average query processing time, etc. and can be used for the logical and technical design and optimal resources sharing in single-level single-mediator *MASs*.

The structure of the paper is the following: Section 2 describes the considered class of *MASs* and provides a formalization in terms of *QT*. Conditions for the existence of the stationary distribution of the queueing network that models *MASs* are given in Section 3, while performance measures and the guidelines for their calculation are listed in Section 4. Finally, we conclude the paper in Section 5.

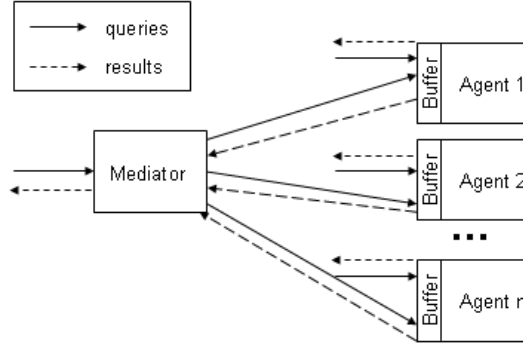
2 Mathematical Model

The class of *MASs* we consider in this paper is in a way similar to the models considered by Zhang and Lesser in [4] and by Horling and Lesser in [1]. The considered *MAS* is of the following structure: the single mediator serves as dispatcher for n independent heterogeneous agents which handle user queries. The queries are propagated just from the mediator to agents, therefore we are dealing with a single-level system. If there are i free agents at the moment of query arrival, then the query is assigned to $\min\{i, m\}$, $1 \leq m \leq n$, of agents that process it independently. If all agents are busy at the moment of query arrival, then the query is stored in a buffer and can be picked up later, according to the First In - First Out (*FIFO*) discipline, when some agents become free. We assume that an agent can also process queries coming from different sources, e.g. from other *MASs* or from other agents that were not able to process query on their own. The structure of the considered *MAS* system is given in Figure 1(a) and the corresponding queueing network is represented in Figure 1(b).

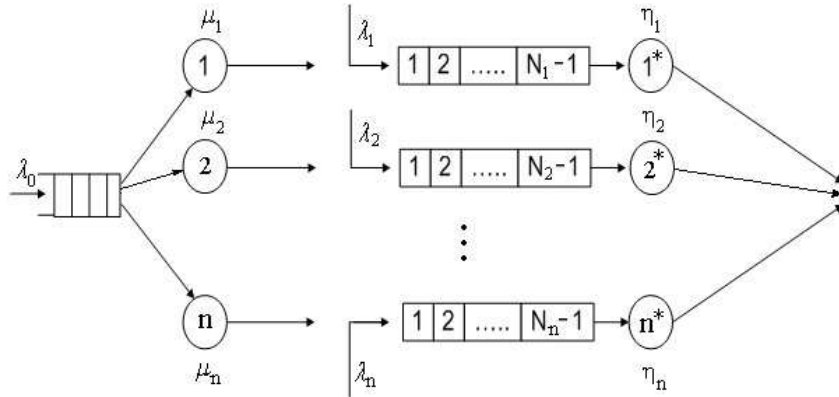
The mediator takes care of queries arriving to the *MAS* and buffers queries in the case of agents' unavailability. The service (query processing) in the *MAS* is performed by agents. Each agent has a finite buffer where queries assigned by the mediator can be stored while agent is busy with another query.

The motivation for the parallel processing of the query by m agents is as follows: (1) agents in the *MAS* can be unreliable, i.e. there is no guarantee that an agent will accomplish the task assigned to him, or an autonomous agent can decline to process the query. Also, the agent can reject the query from mediator just because the capacity of his buffer is exhausted; (2) the results of the query processing by different agents can vary greatly in terms of quality and performance because of the differences in the capabilities of agents, available resources, etc.; (3) in noisy domains the results can be distorted while passing from the agent to the mediator, so it is necessary to wait for results from several agents and to analyze them. In such situation, parallel sending of query to all

⁴ We use the term query to refer to a generic kind of task or service request that comes from outside of the system and requires processing by agents.



(a) The structure of the *MAS* system



(b) Queueing network model for the operation of the considered *MAS* system

Fig. 1. The mathematical model of *MAS*

currently available agents increases chance of an arbitrary query to be successfully processes in the *MAS*. Besides this reliability aspect, parallel handling of query by several agents can decrease response time because the response time in this case is the minimum of durations of handling the query by all involved agents. The results from the field of Queueing Systems [7] have shown that this kind of assumptions is reasonable and allows for achieving a higher performance (comparing with standard one query-one agent systems) in case the arrival rate of the queries is not very high.

As an example of the described *MAS* we can consider a multi-agent information retrieval system, where agents have heterogeneous datasources and therefore process queries in different ways with different performance. Another example is a university network where computers can be used for distributed computations. In such a network, we cannot make an a priori estimation of the time required for the query processing by a single computer, because at each moment a user

can start utilizing the computer therefore making assigned computations slower or, even, canceling them.

The queueing network corresponding to the MAS consists of two interacting parts (Figure 1(b)). In the sequel, the left part of the network (mediator part) will be referred to as the queueing system number 0. It consists of one buffer with an infinite capacity and n possibly heterogeneous servers (links to agents). We refer to these servers as server number $1, \dots, n$. The right part (autonomous agents part) consists of n independent service systems (agents) referred below to as the queueing system number $1^*, \dots, n^*$. Each of these systems has a finite buffer and a single server (agent). The capacity of the buffer of the system number k^* is equal to $N_{k^*} - 1$, so the maximal number of queries in this system is equal to $N_{k^*}, k = \overline{1, n}$, where $N_{k^*} = 1$ corresponds to the case when agent can process only one query at a time.

We assume that queries arrive to the queueing system number 0 according to the stationary Poisson process with intensity λ_0 . If i of servers $1, \dots, n$ are idle at the arrival epoch, the query starts the service in $\min\{i, m\}$ of these servers simultaneously. The discipline of choosing concrete servers, e.g. the fastest available, random, etc. should be specified additionally. Here we consider an arbitrary discipline. We assume that service times in the servers are mutually independent random variables having exponential distribution with parameter μ_k for the server number $k, k = \overline{1, n}$. If all servers $1, \dots, n$ are busy at the arrival epoch, the query goes to the buffer of the queueing system number 0. We assume that this buffer has an infinite capacity. The queries are picked up from the buffer when any of servers $1, \dots, n$ completes the service of previous queries according to the *FIFO* discipline.

After the service in the server k , the query moves for the service in the queueing system number $k^*, k = \overline{1, n}$. If the server of that system (agent of MAS) is idle at the arrival epoch, it starts processing of the arriving query with probability $q_k^{(1)}$ or declines the offer to serve this query with probability $1 - q_k^{(1)}$. Service times of successive queries in the server k^* are independent random variables distributed exponentially with parameter $\eta_k, k = \overline{1, n}$. After the service, query leaves the system number k^* and the network.

If the server k^* is busy at a query arrival epoch from the queueing system number 0, the arriving query with probability $1 - q_k^{(2)}$ is rejected and with supplementary probability it should be placed into the buffer of capacity $N_{k^*} - 1, k = \overline{1, n}$. If the buffer is already full at arrival epoch, the query is lost in the queueing system number k^* .

Besides processing the queries from the queueing system number 0, the server of the system number k^* can also process other queries. These queries arrive to server k^* according to the stationary Poisson process with intensity $\lambda_k, k = \overline{1, n}$. Service times of these queries are distributed exponentially with parameter η_k . In the case the buffer is full at the arrival epoch, the query is lost. No priority for any kind of queries is assigned.

Thus, operation of the queueing network presented in Figure 1(b) is completely described. Our purpose is to perform the stationary analysis of distri-

bution of the number of queries in the nodes of this queueing network and computing its main performance measures.

3 Stationary state distribution of the network

Behavior of the queueing network under study can be described by the multi-dimensional continuous time Markov chain

$$\xi_t = \{j_t, i_t^{(1)}, \dots, i_t^{(k)}\}, t \geq 0, i_t^{(k)} = \overline{0, N_k}, k = \overline{1, n},$$

where the component $i_t^{(k)}$ is equal to the number of queries in queueing system k^* , $k = \overline{1, n}$, at the moment $t, t \geq 0$. It includes the queries in the corresponding buffer, if any, and the query in the server. Component j_t describes the state of the n -server queueing system number 0. The state $j, j \geq 1$, of the component j_t corresponds to the state of the queueing system number 0 when there are j query in a buffer (sure, all the servers of this system are busy).

If the queue in this system is absent, the state of the component j_t is described by the group of n numbers $\{l_1, \dots, l_n\}$ where the entry l_k has value 0 if the k th server is idle and value 1 if the k th server is busy at epoch $t, t \geq 0$. We denote the set of all such states by \mathcal{L} . It is evident that it consists of 2^n states.

Aiming to simplify denotations and use benefits of the matrix analytic methods, we enumerate the components of the process $\xi_t = \{j_t, i_t^{(1)}, \dots, i_t^{(n)}\}, t \geq 0$, in the lexicographic order. Then, we refer to the whole set of states $\{j, i_t^{(1)}, \dots, i_t^{(n)}\}, i_t^{(k)} = \overline{0, N_k}, k = \overline{1, n}$, as to the state j of the process $\xi_t, t \geq 0$,

$$j = \{\underbrace{0, \dots, 0}_n\}, \{\underbrace{0, \dots, 0, 1}_n\}, \dots, \{\underbrace{1, \dots, 1}_n\}, 1, 2, 3, \dots$$

For use in the sequel, we introduce the following notation. $\mu = \sum_{k=1}^n \mu_k$; I is

identity matrix of dimension $K = \prod_{k=1}^n (N_k + 1)$; I_k is identity matrix of dimension

$N_k + 1, k = \overline{1, n}$; O is zero square matrix of dimension K ; $O_{l,m}$ is zero matrix of dimension $Kl \times Km$; \otimes is the symbol of Kronecker product of the matrices; \oplus is the symbol of Kronecker sum of the matrices; T denotes transposition of a matrix or vector; \mathbf{e}_k is the column vector of dimension $N_k + 1$ consisting of all 1's; \mathbf{e}_K is the column vector of dimension K consisting of all 1's; $\mathbf{0}_k$ is the row vector of dimension $N_k + 1$ consisting of all 0's; $k = \overline{1, n}$; $\mathbf{0}_K$ is the row vector of dimension K consisting of all 0's; $\mathbf{f}_k^{(i)}$ is the column vector of dimension $N_k + 1$ having the form $(\underbrace{0, \dots, 0}_i, 1, 0, \dots, 0)^T, i = \overline{0, N_k}, k = \overline{1, n}$;

$\tilde{\mathbf{e}}_k^{(i)}, i = \overline{0, N_k}$, is the column vector of dimension K defined by formula $\tilde{\mathbf{e}}_k^{(i)} = \mathbf{e}_1 \otimes \dots \otimes \mathbf{e}_{k-1} \otimes \mathbf{f}_k^{(i)} \otimes \mathbf{e}_{k+1} \otimes \dots \otimes \mathbf{e}_n$; $\tilde{I}_k, \tilde{I}_k^+, \tilde{I}_k^-, I_k^0$ are the square matrices

of dimension $N_k + 1$, $k = \overline{1, n}$, having the following structure:

$$\tilde{I}_k = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \hat{I}_k = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, I_k^+ = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

Here \tilde{I}_k is obtained from an identity matrix by replacing the entry in the last row and column with 0; \hat{I}_k is obtained from an identity matrix by replacing the entry in the first row and column with 0; I_k^+ is the matrix having 1's in the first over-diagonal and all all other entries equal to 0, $I_k^- = (I_k^+)^T$, $I_k^0 = I_k - \hat{I}_k$; $J_k = q_k^{(1)} I_k^0 I_k^+ + (1 - q_k^{(1)}) I_k^0 I_k + q_k^{(2)} \hat{I}_k I_k^+ + (1 - q_k^{(2)}) \hat{I}_k I_k$, $\mathcal{A}_k = \lambda_k \tilde{I}_k + \eta_k \hat{I}_k$; $\mathcal{A} = \bigoplus_{k=1}^n \mathcal{A}_k = \mathcal{A}_1 \oplus \dots \oplus \mathcal{A}_n$; $\mathcal{B}_k = \lambda_k I_k^+ + \eta_k I_k^-$; $\mathcal{B} = \bigoplus_{k=1}^n \mathcal{B}_k$; $\mathcal{H} = \mathcal{B} - \mathcal{A}$; $\mathcal{C}_k = I_1 \otimes \dots \otimes I_{k-1} \otimes \mu_k J_k \otimes I_{k+1} \otimes \dots \otimes I_n$; $\mathcal{C} = \sum_{k=1}^n \mathcal{C}_k$; $\mathcal{M}_k = I_1 \otimes \dots \otimes I_{k-1} \otimes \mu_k I_k \otimes I_{k+1} \otimes \dots \otimes I_n$; $\mathcal{E} = \sum_{k=1}^n \mathcal{M}_k$;

$$\mathcal{D}_0 = \lambda_0 I, \mathcal{D}_1 = -\lambda_0 I - \mathcal{E} + \mathcal{H}, \mathcal{D}_2 = \mathcal{C};$$

$$Q_{1,0} = (O_{1,2^{n-1}} \mathcal{D}_2), Q_{0,1} = \begin{pmatrix} O_{2^{n-1},1} \\ \mathcal{D}_0 \end{pmatrix}.$$

Let $Q_{0,0}$ be the blocking matrix consisting of matrices

$$Q^{\{l_1, \dots, l_n\}, \{l'_1, \dots, l'_n\}}, \{l_1, \dots, l_n\}, \{l'_1, \dots, l'_n\} \in \mathcal{L},$$

which are defined via introduced matrices \mathcal{H} , \mathcal{C}_k , \mathcal{M}_k depending on the discipline adopted to choose agents for the query processing. The matrix $Q_{0,0}$ can be decomposed as

$$Q_{0,0} = \begin{pmatrix} \tilde{Q}_{0,0} & \tilde{\mathcal{D}}_0 \\ V & \mathcal{D}_0 \end{pmatrix}$$

Denote by Q the block matrix which is the generator of the Markov chain $\xi_t, t \geq 0$.

Lemma: The generator Q has the following block structure:

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & O_{2^n,1} & O_{2^n,1} & O_{2^n,1} & O_{2^n,1} & \dots \\ Q_{1,0} & \mathcal{D}_1 & \mathcal{D}_0 & O & O & O & \dots \\ O_{1,2^n} & \mathcal{D}_2 & \mathcal{D}_1 & \mathcal{D}_0 & O & O & \dots \\ O_{1,2^n} & O & \mathcal{D}_2 & \mathcal{D}_1 & \mathcal{D}_0 & O & \dots \\ O_{1,2^n} & O & O & \mathcal{D}_2 & \mathcal{D}_1 & \mathcal{D}_0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

Theorem 1. Stationary distribution of the Markov chain $\xi_t, t \geq 0$, exists if and only if the following inequality holds true: $\lambda_0 < \sum_{k=1}^n \mu_k$.

Let us denote the stationary probabilities of the states of the Markov chain $\xi_t, t \geq 0$, by

$$p(j, i_1, \dots, i_n) = \lim_{t \rightarrow \infty} P\{j_t = j, i_t^{(1)} = i_1, \dots, i_t^{(n)} = i_n\},$$

$$j = \{l_1, \dots, l_n\} \in \mathcal{L}, 1, 2, \dots; i_k = \overline{0, N_k}, k = \overline{1, n}.$$

According to the lexicographic enumeration of the components of the Markov chain $\xi_t, t \geq 0$, which was already exploited above, we combine probabilities $p(j, i_1, \dots, i_n)$, $i_k = \overline{0, N_k}, k = \overline{1, n}$, into probability row vectors $\mathbf{p}_j, j = \{l_1, \dots, l_n\} \in \mathcal{L}, 1, 2, \dots$ and the macro-vector $\mathbf{p} = (\mathbf{p}_{\{0\}}, \mathbf{p}_1, \mathbf{p}_2, \dots)$ where $\mathbf{p}_{\{0\}} = (\mathbf{p}_{\{0, \dots, 0\}}, \mathbf{p}_{\{0, \dots, 0, 1\}}, \dots, \mathbf{p}_{\{1, \dots, 1\}})$.

Theorem 2. Stationary probability vectors $\mathbf{p}_{\{l_1, \dots, l_n\}}, \{l_1, \dots, l_n\} \in \mathcal{L}$, $\mathbf{p}_1, \mathbf{p}_2, \dots$ are calculated in the following way:

- the vector $\mathbf{p}_{\{l_1, \dots, l_n\}}$ is computed as the block number $\sum_{k=1}^n l_k 2^{n+1-k} + 1$ in the block vector $\mathbf{p}_{\{1, \dots, 1\}} \mathcal{F}_1$, $\{l_1, \dots, l_n\} \in \mathcal{L}$, $\{l_1, \dots, l_n\} \neq \{1, \dots, 1\}$;
- the vectors $\mathbf{p}_j, j \geq 1$, are computed by $\mathbf{p}_i = \mathbf{p}_{\{1, \dots, 1\}} R^i, i \geq 1$, where $\mathcal{F}_1 = -V(\tilde{Q}_{0,0})^{-1}$, $\mathcal{F} = \mathcal{D}_1 + \mathcal{F}_1 \tilde{D}_0$;
- the matrix R is a minimal non-negative solution to the matrix equation

$$R^2 \mathcal{D}_2 + R \mathcal{D}_1 + \mathcal{D}_0 = O;$$

- the vector $\mathbf{p}_{\{1, \dots, 1\}}$ is the unique solution to the following system of linear algebraic equations

$$\mathbf{p}_{\{1, \dots, 1\}} [\mathcal{F} + R \mathcal{D}_2] = \mathbf{0}_K, \mathbf{p}_{\{1, \dots, 1\}} [\mathcal{F}_1 + (I - R)^{-1}] \mathbf{e}_K = 1.$$

This theorem gives a straightforward easily-implementable algorithmic way for the calculation of the stationary probability vector \mathbf{p} .

4 Calculation of the Network Performance Measures

Having the stationary probability vectors been computed, we can calculate different performance measures of the queueing network. Formulae for calculation of some of them are given below.

Stationary distribution of the number of queries in the system k^* is given by the vector $\boldsymbol{\theta}^{(k)}$ having components

$$\boldsymbol{\theta}_i^{(k)} = \mathbf{p}_{\{1, \dots, 1\}} [\mathcal{F}_1 + (I - R)^{-1}] \tilde{\mathbf{e}}_k^{(i)}, i = \overline{0, N_k}, k = \overline{1, n}.$$

Average number of queries L_k in the system k^* , average number of queries L_0 in the system number 0 and average total number of queries L in the network are calculated by

$$L_0 = \sum_{\{l_1, \dots, l_n\} \in \mathcal{L}} \sum_{j=1}^n l_j \mathbf{p}_{\{l_1, \dots, l_n\}} \mathbf{e}_K + \mathbf{p}_{\{1, \dots, 1\}} ((n+1)I - nR) R (I - R)^{-2} \mathbf{e}_K.$$

$$L_k = \sum_{i=1}^{N_k} i \theta_i^{(k)}, k = \overline{1, n}, L = \sum_{k=0}^n L_k.$$

Probabilities $P_{loss}^{(k)}$ that an arbitrary query arriving to the system k^* will be rejected due to desire of agent or because the buffer is full and is calculated by the following formula:

$$P_{loss}^{(k)} = \theta_{N_k}^{(k)} + (1 - q_1^{(k)}) \theta_0^{(k)} + (1 - q_2^{(k)}) (1 - \theta_0^{(k)} - \theta_{N_k}^{(k)}), k = \overline{1, n}.$$

Probability P_{loss} that an arbitrary query arriving to the *MAS* will not get service by any agent is computed by formula

$$P_{loss} = \mathbf{p}_{\{1, \dots, 1\}} (I + R(I - R)^{-1}) \mathbf{e}_K \sum_{k=1}^n \frac{\mu_k}{\mu} P_{loss}^{(k)} + \\ + \sum_{\{l_1, \dots, l_n\} \in \mathcal{L}, (k_1, \dots, k_{\hat{m}})} \mathbf{p}_{\{l_1, \dots, l_n\}} \mathbf{e}_K \mathcal{B}_{\{l_1, \dots, l_n\}}^{k_1, \dots, k_{\hat{m}}} (1 - \prod_{r=1}^{\hat{m}} (1 - P_{loss}^{(k_r)})),$$

where $\hat{m} = \min\{m, n - l_1 - \dots - l_n\}$, $\mathcal{B}_{\{l_1, \dots, l_n\}}^{k_1, \dots, k_{\hat{m}}}$ is probability of assigning the agents number $k_1, \dots, k_{\hat{m}}$ for service providing to an arbitrary query which arrives when the states of servers are defined by the set $\{l_1, \dots, l_n\}$. This probability is easily computed when strategy of agents assigning is fixed.

Average sojourn time $\tilde{W}_1^{(0)}$ in the system number 0 and sojourn time $\tilde{W}_1^{(k)}$ for queries that are not rejected in the system number k^* are calculated as follows:

$$\tilde{W}_1^{(0)} = \lambda_0^{-1} \tilde{L}_0, \tilde{W}_1^{(k)} = \sum_{i=0}^{N_k-1} \frac{i+1}{\eta_k} \theta_i^{(k)}, \tilde{W}_1^{(k)} = \frac{\tilde{W}_1^{(k)}}{1 - P_{loss}^{(k)}}, k = \overline{1, n}.$$

Average sojourn time V of a query in the queueing network is computed by

$$V = \tilde{W}_1^{(0)} + \mathbf{p}_{\{1, \dots, 1\}} (I + R(I - R)^{-1}) \mathbf{e}_K \sum_{k=1}^n \frac{\mu_k}{\mu} \tilde{W}_1^{(k)} + \\ + \sum_{(\{l_1, \dots, l_n\}) \in \mathcal{L}} \mathbf{p}_{\{l_1, \dots, l_n\}} \mathbf{e}_K \mathcal{B}_{\{l_1, \dots, l_n\}}^{k_1, \dots, k_{\hat{m}}} W(k_1, \dots, k_{\hat{m}}),$$

$W(k_1, \dots, k_{\hat{m}})$ is expectation of minimum of sojourn times in systems number $k_1, \dots, k_{\hat{m}}$.

5 Conclusion

We have analyzed the process of user query processing in a particular class of *MASs* in terms of the queueing network. Tree-like structure of the network topology allows to get the steady state-distribution of the network states in the

exact analytic form. Main performance measures of the network are calculated and can be used for the quantitative analysis of a particular *MAS*. For instance, having specified the system parameters, it is possible to calculate the average time of query processing in the system, or the probability of query being rejected, etc. The results are extendable to the cases where the input and service processes have more complicated nature. Modifications to the considered *MAS*, where the results of query processing are unreliable because of errors or because the agents are subject to breakdowns and recovering be can investigated analogously.

6 Acknowledgments

This research was supported in part by Ministry of Information and Communication (MIC) Korea, under the IT Foreign Specialist Inviting Program (ITSIP), ITSOC, International Cooperative Research by Ministry of Science and Technology, KOTEF, and 2nd stage Brain Korea 21.

References

1. Horling B., Lesser V.: Using Queueing Theory to Predict Organizational Metrics. AAMAS'06, May 8-12, 2006, Hakodate, Japan. ACM Press. 1098-1100.
2. Gnanasambandam N., Lee S.C, Gautam N., et al.: Reliable MAS Performance Prediction Using Queueing Models. IEEE First Symposium on Multi-Agent Security and Survivalibility, 2004. 55-64.
3. Gnanasambandam N.: Survivability of Multi-Agent Systems. AAMAS'05 July, 25-29, 2005, Utrecht, Netherlands. ACM Press. 1376.
4. Zhang H., Lesser V.: A Dynamically Formed Hierarchical Agent Organization for a Distributed Content Sharing System . IAT 2004, September 2004, Beijing. 169175.
5. Okamoto S., Scerri P., Sycara K.: Toward an Understanding of the Impact of Software Personal Assistants on Human Organizations. AAMAS'06, May 8-12, 2006, Hakodate, Japan. ACM Press. 630-637.
6. Vazques-Salceda J., Dignum V., Dignum F.: Organizing Multiagent Systems. Autonomous Agents and Multi-Agent Systems, 2005, 11, 307-360.
7. Lee M.H., Dudin A.N., Klimenok V.I. The SM/M/N queueing system with broadcasting service. Mathematical Problems in Engineering. 2006. Article ID 98171.
8. Horling B., Lesser V.: A survey of multi-agent organizational paradigms. Knowledge Engineering Review, Cambridge University Press, 2004, 19, 281-316.