

# СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 519.872, 519.21

## СЕТИ МАССОВОГО ОБСЛУЖИВАНИЯ ПРОИЗВОЛЬНОЙ ТОПОЛОГИИ С ДЕЛЕНИЕМ И СЛИЯНИЕМ ТРЕБОВАНИЙ: СЛУЧАЙ БЕСКОНЕЧНОПРИБОРНЫХ СИСТЕМ ОБСЛУЖИВАНИЯ

Осипов О.А., Тананко И.Е.  
СГУ имени Н.Г. Чернышевского, г. Саратов

---

*Поступила в редакцию 29.06.2017, после переработки 11.12.2017.*

---

В работе рассматривается класс открытых сетей массового обслуживания произвольной топологии, который является развитием классических fork-join сетей массового обслуживания, используемых в качестве математических моделей стохастических систем с параллельным и распределенным принципом функционирования (GRID-системы, RAID-массивы, MapReduce и т. д.)

Системы обслуживания в рассматриваемой сети поделены на три типа в зависимости от их назначения: бесконечноприборные базовые системы, дивайдеры, интеграторы. Наличие бесконечного числа обслуживающих приборов в базовых системах позволяет существенно упростить анализ и рассмотреть сети обслуживания с произвольной топологией. Требование, поступающее в дивайдер, делится на некоторое число частей — фрагментов. Полученные фрагменты обслуживаются независимо друг от друга в базовых системах сети, переходят по сети. Каждый из фрагментов может снова поделиться на фрагменты при поступлении в дивайдер. Объединение фрагментов происходит в интеграторах. Так, перед уходом из сети все фрагменты объединяются в одном из интеграторов в исходное требование.

Главным результатом работы стал метод получения распределения длительности пребывания требований в изучаемой сети массового обслуживания.

**Ключевые слова:** сети массового обслуживания с делением и слиянием требований, сети массового обслуживания, длительность пребывания, распределенная обработка, фазовое распределение.

*Вестник ТвГУ. Серия: Прикладная математика. 2017. № 4. С. 43–58.*  
<https://doi.org/10.26456/vtprm188>

### Введение

Сети массового обслуживания (SeMO) с делением и слиянием требований (fork-join queueing networks) являются математическими моделями, используемы-

ми для анализа дискретных стохастических систем с сетевой структурой и параллельным принципом функционирования (телекоммуникационные системы, распределенные базы данных, многопроцессорные системы, GRID-системы) [1].

Ключевой особенностью таких систем является деление поступающих для выполнения запросов на несколько подзапросов, которые обслуживаются параллельно. После полного выполнения подзапросов поступившего запроса они синхронизируются — происходит их объединение, а запрос считается выполненным.

В работе [2] сеть массового обслуживания с делением и слиянием требований применяется для анализа технологии распределенных вычислений MapReduce и метода маршрутизации Multipath Routing. Результаты по моделированию RAID-массивов и распределенных баз данных представлены в [1, 3]. Обсуждению систем облачных вычислений и анализу параллельных алгоритмов посвящены, например, статьи [4, 5]. Однако, несмотря на обширное применение сетей обслуживания с делением и слиянием требований, в опубликованных работах рассматриваются в основном сети параллельных систем массового обслуживания (СМО). Для сети обслуживания, состоящей из двух одноприборных параллельных СМО, в статье [6] получено выражение для производящей функции стационарного распределения вероятностей состояний сети. Анализ же сетей обслуживания большей размерности с делением и слиянием требований проводится только приближенными методами [7–10]. Обзор всех основных теоретических и прикладных результатов за тридцатилетний период изучения СеМО с делением и слиянием требований можно найти в [1].

В данной работе будут рассмотрены сети массового обслуживания с делением и слиянием требований произвольной топологии. Первая часть работы посвящена описанию рассматриваемой сети обслуживания. Во второй части представлен метод анализа сети обслуживания. Численный пример представлен в третьей части. Результаты работы и возможные направления дальнейших исследований обсуждаются в заключении.

## 1. Описание сети массового обслуживания

Рассматривается открытая сеть массового обслуживания  $\mathcal{M}$  с одним классом требований и множеством  $\{N_1, \dots, N_L\}$  систем массового обслуживания. В сеть из источника  $N_0$  поступает пуассоновский поток требований с интенсивностью  $\Lambda_0$ . Любое из требований может многократно делиться на части — фрагменты. Полученные фрагменты обслуживаются независимо друг от друга, переходят по системам сети обслуживания. Каждый из фрагментов может снова разделиться на новые фрагменты. Перед уходом из сети все фрагменты должны быть снова объединены в исходное требование.

Далее будем использовать термин «фрагмент» для описания и требования, и его частей.

### 1.1 Типы систем обслуживания

Пусть  $X = \{1, \dots, L\}$  — множество номеров систем обслуживания. Системы в сети обслуживания  $\mathcal{M}$  поделены на три непустых, непересекающихся множества в соответствии с их номерами:

1.  $X_S = \{x_1^S, \dots, x_M^S\}$  — множество номеров базовых систем;
2.  $X_{\mathcal{F}} = \{x_1^{\mathcal{F}}, \dots, x_K^{\mathcal{F}}\}$  — множество номеров дивайдеров;
3.  $X_{\mathcal{J}} = \{x_1^{\mathcal{J}}, \dots, x_K^{\mathcal{J}}\}$  — множество номеров интеграторов.

Заметим, что число дивайдеров  $K$  совпадает с числом интеграторов, а  $X = X_S \cup X_{\mathcal{F}} \cup X_{\mathcal{J}}$ . Обозначим через

- $S_i$  — базовая система  $N_{x_i^S}$ ,  $i = 1, \dots, M$ ;
- $F_j$  — дивайдер  $N_{x_j^{\mathcal{F}}}$ ,  $j = 1, \dots, K$ ;
- $J_k$  — интегратор  $N_{x_k^{\mathcal{J}}}$ ,  $k = 1, \dots, K$ .

Системы обслуживания, принадлежащие одному множеству, имеют сходное назначение и особенности.

1.  $\mathcal{S} = \{N_{x_1^S}, \dots, N_{x_M^S}\} = \{S_1, \dots, S_M\}$  — множество базовых систем обслуживания — множество систем массового обслуживания с бесконечным числом одинаковых обслуживающих приборов. Длительность обслуживания фрагмента на любом приборе базовой системы  $S_i$  имеет экспоненциальное распределение с параметром  $\mu_i$ ,  $i = 1, \dots, M$ .
2.  $\mathcal{F} = \{N_{x_1^{\mathcal{F}}}, \dots, N_{x_K^{\mathcal{F}}}\} = \{F_1, \dots, F_K\}$  — множество дивайдеров — множество одноприборных систем массового обслуживания. Процесс обслуживания в дивайдере  $F_k$ ,  $k = 1, \dots, K$ , заключается в мгновенном делении поступающего фрагмента на фиксированное число  $d_k \geq 2$  новых фрагментов. Каждый из  $d_k$  фрагментов мгновенно покидает дивайдер.
3.  $\mathcal{J} = \{N_{x_1^{\mathcal{J}}}, \dots, N_{x_K^{\mathcal{J}}}\} = \{J_1, \dots, J_K\}$  — множество интеграторов — множество систем массового обслуживания с бесконечным числом приборов. Процесс обслуживания в интеграторе заключается в следующем. Каждый поступающий фрагмент занимает свободный прибор и ожидает момента поступления в интегратор последнего из всех фрагментов, принадлежащих до разделения тому же самому ранее единому фрагменту. Сразу после этого эти фрагменты мгновенно освобождают обслуживающие приборы, объединяются в исходный фрагмент, который покидает интегратор.

## 1.2 Сигнатура фрагмента

Опишем формально процессы, происходящие с фрагментами в системах сети обслуживания. Каждый из фрагментов в сети обслуживания характеризуется сигнатурой, которая идентифицирует фрагмент и отражает связи с другими фрагментами.

**Определение 1.** Пусть  $\sigma$  — некоторый произвольный фрагмент в сети обслуживания, который получен непосредственно в результате деления фрагмента  $\sigma^*$  в дивайдере  $F_k$ . Пусть  $i$  — номер фрагмента  $\sigma$  при делении, то есть  $i \in \{1, \dots, d_k\}$ . Сигатурой фрагмента  $\sigma$  назовем тройку  $(\sigma^*, k, i)$ . В том случае, когда  $\sigma$  является поступающим из источника требованием, сигатура имеет вид  $(\sigma, 0, 1)$ .

Если фрагмент  $\sigma$  имеет сигнатуру  $(\sigma^*, k, i)$ , то будем записывать это  $\sigma \sim (\sigma^*, k, i)$ .

Предположим, что фрагмент  $\sigma$  поступает в произвольный дивайдер  $F_k$ ,  $k \in \{1, \dots, K\}$ , который делит фрагмент  $\sigma$  на  $d_k$  фрагментов  $\{\sigma_1, \dots, \sigma_{d_k}\}$ ,  $\sigma_i \sim (\sigma, k, i)$ ,  $i = 1, \dots, d_k$ . Полученные в результате деления фрагменты мгновенно покидают дивайдер и поступают в системы обслуживания. Базовые системы не меняют сигнатуру фрагмента, а каждый из полученных фрагментов может снова разделиться на новые фрагменты при поступлении в другой дивайдер. Для корректного объединения фрагментов  $\{\sigma_1, \dots, \sigma_{d_k}\}$  они все должны в итоге поступить в некоторый интегратор. Поэтому потребуем выполнения следующего условия: маршрутизация фрагментов должна быть задана так, что все фрагменты  $\sigma_i$ ,  $i = 1, \dots, d_k$ , полученные в дивайдере  $F_k$ , должны впоследствии поступить и быть объединены только в соответствующем интеграторе  $J_k$ . Переход данных фрагментов в интеграторы  $J_l$ ,  $l \neq k$ , не допускается.

При поступлении фрагмента  $\sigma_i \in \{\sigma_1, \dots, \sigma_{d_k}\}$  в интегратор  $J_k$  он находится в этом интеграторе до тех пор, пока не поступят все оставшиеся фрагменты из данного множества. Как только это произошло, фрагменты  $\{\sigma_1, \dots, \sigma_{d_k}\}$  освобождают обслуживаемые их приборы, мгновенно объединяются в исходный фрагмент  $\sigma$ , который сразу же покидает интегратор, переходя в другую систему обслуживания или источник.

Для удобства фрагмент  $\sigma$ , характеризуемый сигатурой  $(\sigma^*, k, i)$ , будем называть  $k$ -фрагментом. Напомним, что фрагмент может вернуться в источник требований  $N_0$  только в случае, когда он является требованием (0-фрагментом).

### 1.3 Маршрутизация фрагментов

Для каждого фрагмента в сети обслуживания введем в рассмотрение вектор перемещений  $v$ .

**Определение 2.** Пусть произвольный фрагмент  $\sigma$  характеризуется сигатурой  $(\sigma_m, k_m, i_m)$ , и при этом  $\sigma_n \sim (\sigma_{n-1}, k_{n-1}, i_{n-1})$ ,  $n = m, m-1, \dots, 2$ ,  $\sigma_1 \sim (\sigma_1, 0, 1)$ . Вектор  $v = (k_0, k_1, \dots, k_m) = (0, k_1, \dots, k_m)$  назовем вектором перемещений для  $k_m$ -фрагмента  $\sigma$ , где  $k_m$  — ведущий элемент вектора перемещений.

Полагаем, что маршрутизация фрагментов такова, что вектор перемещений каждого фрагмента не может содержать одинаковых элементов, то есть  $k_i \neq k_j$ ,  $i \neq j$ ,  $i, j = 1, \dots, m$ . Это условие запрещает повторное деление фрагментов, полученных ранее в результате деления в дивайдере  $F_k$ , снова в этом дивайдере без их предварительного объединения в единый фрагмент в интеграторе  $J_k$ .

Все векторы перемещений тогда образуют множество допустимых векторов перемещений  $\mathcal{V}$ , которое является конечным множеством, зависящим от топологии

сети обслуживания. Пусть  $\mathbf{v} = (0, k_1, \dots, k_m)$ , тогда через  $(\mathbf{v}, k_{m+1})$  обозначим вектор  $(0, k_1, \dots, k_m, k_{m+1})$ .

Возможные переходы фрагментов в сети обслуживания  $\mathcal{N}$  заданы при помощи набора матриц передач  $\Theta = (\Theta^{(0)}, \Theta^{(1)}, \dots, \Theta^{(K)})$ , где  $\Theta^{(k)} = (\theta_{i,j}^{(k)})$ ,  $i, j = 0, 1, \dots, L$ ,  $k = 0, 1, \dots, K$ .

Элемент  $\theta_{i,j}^{(k)}$  определяет

- вероятность перехода  $k$ -фрагмента из  $N_i$  в  $N_j$ , если  $N_i \notin \mathcal{F}$ ;
- число  $k$ -фрагментов, полученных при делении фрагмента в дивайдере  $N_i$ , которые перейдут в  $N_j$ , если  $N_i \in \mathcal{F}$ .

Очевидно, что тогда

$$\theta_{x_{\mathcal{F}},j}^{(k)} = 0, \quad i = 1, \dots, K, i \neq k, j = 0, 1, \dots, L;$$

$$\sum_{j=0}^L \theta_{x_{\mathcal{F}},j}^{(k)} = d_k.$$

Предполагается, что набор матриц передач  $\Theta$  обеспечивает корректную маршрутизацию фрагментов, то есть маршрутизация заданная набором  $\Theta$  удовлетворяет условиям заданным выше.

## 2. Анализ сети обслуживания

### 2.1 Потоки в сети обслуживания

Обозначим через  $\lambda_{in}^{(\mathbf{v})}(N_i)$ ,  $\lambda_{out}^{(\mathbf{v})}(N_i)$  интенсивности входящего и выходящего потоков фрагментов с вектором перемещений  $\mathbf{v} \in \mathcal{V}$  для системы обслуживания  $N_i$ ,  $i = 1, \dots, L$ . Тогда суммарные интенсивности входящего  $\Lambda_{in}(N_i)$  и выходящего  $\Lambda_{out}(N_i)$  потоков фрагментов для систем обслуживания  $N_i$ ,  $i = 1, \dots, L$ ,

$$\Lambda_{in}(N_i) = \sum_{\mathbf{v} \in \mathcal{V}} \lambda_{in}^{(\mathbf{v})}(N_i), \tag{1}$$

$$\Lambda_{out}(N_i) = \sum_{\mathbf{v} \in \mathcal{V}} \lambda_{out}^{(\mathbf{v})}(N_i). \tag{2}$$

Пусть  $k$  — ведущий элемент вектора перемещений  $\mathbf{v} \in \mathcal{V}$ , в стационарном режиме функционирования сети обслуживания интенсивности входящих и выходящих потоков связаны следующими соотношениями (3), (4):

$$\lambda_{in}^{(0)}(N_j) = \Lambda_0 \theta_{0,j}^{(0)} + \sum_{i \in X_S \cup X_{\mathcal{F}}} \lambda_{out}^{(0)}(N_i) \theta_{i,j}^{(0)}, \quad k = 0; \tag{3}$$

$$\lambda_{in}^{(\mathbf{v})}(N_j) = \sum_{i \in X_S \cup X_{\mathcal{F}}} \lambda_{out}^{(\mathbf{v})}(N_i) \theta_{i,j}^{(k)} + \frac{1}{d_k} \lambda_{out}^{(\mathbf{v})}(F_k) \theta_{x_{\mathcal{F}},j}^{(k)}, \quad k > 0; \tag{4}$$

где  $j = 1, \dots, L$ .

Справедливо, что

$$\lambda_{out}^{(v)}(S_i) = \lambda_{in}^{(v)}(S_i), \quad i = 1, \dots, M, \quad (5)$$

$$\lambda_{out}^{(v,k)}(F_k) = d_k \lambda_{in}^{(v)}(F_k), \quad \lambda_{out}^{(v)}(J_k) = \frac{1}{d_k} \lambda_{in}^{(v,k)}(J_k), \quad k = 1, \dots, K. \quad (6)$$

$$\lambda_{in}^{(v)}(F_k) = \lambda_{out}^{(v)}(J_k), \quad k = 1, \dots, K. \quad (7)$$

Интенсивности входящих и выходящих потоков фрагментов можно найти для всех  $v \in \mathcal{V}$  как решение уравнений (3), (4) с использованием равенств (5) и (6). Однако, для вычисления интенсивностей потоков фрагментов удобнее использовать следующий алгоритм, который основан на равенстве (7).

---

Алгоритм вычисления интенсивностей потоков фрагментов

---

1. Из уравнений потоков для 0-фрагментов найти  $\lambda_{in}^{(0)}(N_j)$ , для всех  $j \in X_S \cup X_{\mathcal{F}}$ :

$$\lambda_{in}^{(0)}(N_j) = \Lambda_0 \theta_{0,j}^{(0)} + \sum_{i=1}^M \lambda_{in}^{(0)}(S_i) \theta_{x_i^S, j}^{(0)} + \sum_{i=1}^K \lambda_{in}^{(0)}(F_i) \theta_{x_i^{\mathcal{F}}, j}^{(0)}. \quad (8)$$

2. Для всех векторов перемещений  $v \in \mathcal{V}$ , для которых найдены  $\lambda_{in}^{(v)}(N_i)$ , где  $i \in X_S \cup X_{\mathcal{F}}$ , выполнить шаг 3.

Затем перейти к шагу 4.

3. Для всех  $k$  таких, что  $\lambda_{in}^{(v)}(F_k) > 0$ , найти интенсивности потоков фрагментов  $\lambda_{in}^{(v,k)}(N_j)$ , где  $j \in X_S \cup X_{\mathcal{F}}$ , если они не были найдены ранее, используя уравнения:

$$\lambda_{in}^{(v,k)}(N_j) = \lambda_{in}^{(v)}(F_k) \theta_{x_k^{\mathcal{F}}, j}^{(k)} + \sum_{i=1}^M \lambda_{in}^{(v,k)}(S_i) \theta_{x_i^S, j}^{(k)} + \sum_{i=1}^K \lambda_{in}^{(v,k)}(F_i) \theta_{x_i^{\mathcal{F}}, j}^{(k)}. \quad (9)$$

Возврат к шагу 2.

4. Вычислить суммарные входящие интенсивности потоков фрагментов во все системы обслуживания

$$\Lambda_{in}(N_i) = \sum_{v \in \mathcal{V}} \lambda_{in}^{(v)}(N_i), \quad i \in X_S \cup X_{\mathcal{F}}, \quad (10)$$

$$\Lambda_{in}(J_k) = d_k \Lambda_{in}(F_k), \quad k = 1, \dots, K. \quad (11)$$

Конец алгоритма.

2.2 Длительность пребывания требований в сети обслуживания

**Определение 3.** Множество  $\mathcal{H}_k \subset \{N_1, \dots, N_L\}$ ,  $k \in \{1, \dots, K\}$ , таких систем обслуживания, в которые могут поступить фрагменты с вектором перемещений, содержащим  $k$ , будем называть подсетью порожденной дивайдером  $F_k$ . Положим дополнительно, что  $F_k \in \mathcal{H}_k$ .

**Определение 4.** Пусть сеть обслуживания  $\mathcal{N}$  функционирует в стационарном режиме. Обозначим через  $\tau_k$ ,  $k = 1, \dots, K$ , случайную величину, равную длительности интервала времени от момента разделения в дивайдере  $F_k$  произвольного фрагмента до момента объединения всех его частей в интеграторе  $J_k$ . Будем называть  $\tau_k$  длительностью реакции подсети  $\mathcal{H}_k$ .

**Определение 5.** Матрицу  $\Theta^{(k)}$  из набора матриц передач  $\Theta$  назовем элементарной матрицей, если столбцы, соответствующие переходам во все дивайдеры, состоят из нулей, то есть

$$\theta_{i,x_j}^{(k)} = 0, \quad i = 0, 1, \dots, L, \quad j = 1, \dots, K.$$

Конечность множества допустимых векторов перемещений  $\mathcal{V}$  влечет следующее утверждение.

**Утверждение 1.** Среди матриц передачи  $\{\Theta^{(1)}, \dots, \Theta^{(K)}\}$  существует, по крайней мере, одна элементарная матрица.

**Определение 6.** Пусть для некоторого  $k \in \{1, \dots, K\}$ ,  $\Theta^{(k)}$  — элементарная матрица, тогда будем называть подсеть  $\mathcal{H}_k$  элементарной подсетью.

Пусть  $\mathcal{H}_k$ ,  $k \in \{1, \dots, K\}$ , есть элементарная подсеть. Через  $\mathcal{A}(k)$  обозначим множество номеров базовых систем обслуживания, в которые переходят фрагменты непосредственно после деления в дивайдере  $F_k$ , то есть

$$\mathcal{A}(k) = \left\{ j \in \{1, \dots, M\} : \theta_{x_k^F, x_j^S}^{(k)} > 0 \right\}.$$

Далее, пусть  $\theta_{x_k^F, x_i^S}^{(k)} > 0$ , тогда

$$\mathcal{B}(k, S_i) = (b_1, b_2, \dots, b_{c_b}),$$

$$b_1 = i, \quad b_2 < b_3 < \dots < b_{c_b},$$

обозначает упорядоченную последовательность номеров всех базовых систем  $S_{b_j}$ ,  $j = 2, \dots, c_b$ , в которые возможно поступление  $k$ -фрагментов из системы  $S_i$ .

Пусть  $\xi$  есть непрерывная случайная величина с фазовым распределением и параметрами  $(\alpha, \mathbf{A})$ , обозначим это как  $\xi \sim \text{PH}(\alpha, \mathbf{A})$  [11–14]. Известно [12], что если  $\xi \sim \text{PH}(\alpha, \mathbf{A})$  и  $\eta \sim \text{PH}(\beta, \mathbf{B})$  суть независимые случайные величины, то случайная величина  $\max\{\xi, \eta\}$  имеет фазовое распределение  $\text{PH}(\gamma, \mathbf{C})$ , где  $(\gamma, \mathbf{C}) = (\alpha, \mathbf{A}) \vee (\beta, \mathbf{B})$ , а операция “ $\vee$ ” определена следующим образом

$$\gamma = (\alpha \otimes \beta, (1 - \beta \mathbf{1})\alpha, (1 - \alpha \mathbf{1})\beta), \tag{12}$$

$$C = \begin{pmatrix} \mathbf{A} \oplus \mathbf{B} & \mathbf{I} \otimes (-\mathbf{B}\mathbf{1}) & (-\mathbf{A}\mathbf{1}) \otimes \mathbf{I} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{pmatrix}, \quad (13)$$

здесь  $\mathbf{1}$  — вектор-столбец из единиц,  $\oplus$ ,  $\otimes$  — символы операций суммы и произведения Кронекера,  $\mathbf{I}$  — единичная матрица.

**Теорема 1.** Длительность реакции  $\tau_k$  элементарной подсети  $\mathcal{H}_k$ ,  $k \in \{1, \dots, K\}$ , имеет фазовое распределение РН  $(\boldsymbol{\alpha}^{(k)}, \mathbf{A}^{(k)})$ , где

$$(\boldsymbol{\alpha}^{(k)}, \mathbf{A}^{(k)}) = \bigvee_{j \in \mathcal{A}(k)} \left( \bigvee_{l=1}^{\theta_{x_k^S, x_j^S}^{(k)}} (\hat{\boldsymbol{\alpha}}^{(j)}, \hat{\mathbf{A}}^{(j)}) \right), \quad (14)$$

$$\begin{aligned} \hat{\boldsymbol{\alpha}}^{(j)} &= (\hat{\alpha}_n^{(j)}), \quad n = 1, \dots, c_b, \\ \hat{\alpha}_1^{(j)} &= 1, \end{aligned} \quad (15)$$

$$\begin{aligned} \hat{\mathbf{A}}^{(j)} &= (\hat{a}_{n,m}^{(j)}), \quad n, m = 1, \dots, c_b, \\ \hat{a}_{n,m}^{(j)} &= \mu_{b_n} \theta_{n^*, m^*}^{(k)}, \quad n \neq m, \end{aligned} \quad (16)$$

$$\hat{a}_{n,n}^{(j)} = -\mu_{b_n}. \quad (17)$$

Здесь  $(b_1, \dots, b_{c_b}) = \mathcal{B}(k, S_j)$ ,

$$n^* = x_{b_n}^S, \quad m^* = x_{b_m}^S.$$

*Доказательство.* Рассмотрим произвольный фрагмент  $\sigma$ , поступающий в дивайдер  $F_k$ , в котором происходит его деление на  $d_k$  фрагментов  $\{\sigma_1, \dots, \sigma_{d_k}\}$ . Фрагменты переходят по элементарной подсети обслуживания  $\mathcal{H}_k$  так, что поступают в интегратор  $J_k$  для объединения. В процессе перехода фрагментов от  $F_k$  до  $J_k$  они не делятся, так как  $\mathcal{H}_k$  является элементарной подсетью.

Пусть фрагмент  $\sigma_i$ ,  $i \in \{1, \dots, d_k\}$ , переходит из дивайдера  $F_k$  непосредственно в базовую систему  $S_{q_i}$ ,  $q_i \in \mathcal{A}(k)$ . Пусть  $(b_1, \dots, b_{c_b}) = \mathcal{B}(k, S_{q_i})$ . Для удобства введем следующие обозначения:

$$n^* = x_{b_n}^S, \quad m^* = x_{b_m}^S.$$

Процесс перехода фрагмента  $\sigma_i$ , начиная от момента его поступления в базовую систему  $S_{q_i}$  и до момента его поступления в интегратор  $J_k$ , представим как эволюцию поглощающей цепи Маркова с непрерывным временем [15]. Данная цепь Маркова имеет множество состояний  $\{0, 1, \dots, c_b\}$ . Переход цепи в состояние  $n \in \{1, \dots, c_b\}$  соответствует поступлению фрагмента  $\sigma_i$  в базовую систему  $S_{b_n}$ . Длительность пребывания в состоянии  $n$  есть случайная величина с экспоненциальным распределением с параметром  $\mu_{b_n}$ . Цепь переходит из состояния  $n$  в состояние  $m \in \{1, \dots, c_b\}$  с вероятностью  $\theta_{n^*, m^*}^{(k)}$ , которая является вероятностью



перехода фрагмента  $\sigma_i$  из базовой системы  $S_{b_n}$  в систему  $S_{b_m}$  после окончания обслуживания в базовой системе  $S_{b_n}$ .

Состояние цепи  $m = 0$  является поглощающим состоянием. Переход цепи в это состояние отображает поступление фрагмента  $\sigma_i$  в интегратор  $J_k$ . Цепь переходит из состояния  $n \in \{1, \dots, c_b\}$  в состояние  $m = 0$  с вероятностью  $\theta_{n^*, x_k^j}^{(k)}$ .

Таким образом, процесс перехода фрагмента  $\sigma_i$  представлен в виде процесса эволюции цепи Маркова с начальным распределением  $(\hat{\alpha}_0^{(q_i)}, \hat{\alpha}^{(q_i)})$ , где  $\hat{\alpha}_1^{(q_i)} = 1$ .

Генератор  $\hat{Q}^{(q_i)}$  цепи, описывающей переходы фрагмента  $\sigma_i$ , имеет вид

$$\hat{Q}^{(q_i)} = \begin{pmatrix} 0 & \mathbf{0} \\ -\hat{\mathbf{A}}^{(q_i)} \mathbf{1} & \hat{\mathbf{A}}^{(q_i)} \end{pmatrix}, \quad (18)$$

где

$$\hat{\mathbf{A}}^{(q_i)} = \left( \hat{a}_{n,m}^{(q_i)} \right), \quad n, m = 1, \dots, c_b, \quad (19)$$

$$\hat{a}_{n,m}^{(q_i)} = \mu_{b_n} \theta_{n^*, m^*}^{(k)}, \quad n \neq m, \quad (19)$$

$$\hat{a}_{n,n}^{(q_i)} = -\mu_{b_n}. \quad (20)$$

Обозначим через  $\xi_i$  случайную величину, равную длительности интервала времени до поглощения цепи Маркова, тогда, исходя из вероятностной интерпретации фазового распределения,  $\xi_i \sim \text{PH}(\hat{\alpha}^{(q_i)}, \hat{\mathbf{A}}^{(q_i)})$ .

Длительность реакции  $\tau_k$  элементарной подсети  $\mathcal{H}_k$  тогда может быть определена как наибольшая из всех случайных величин  $\xi_i$ ,  $i = 1, \dots, d_k$ . Таким образом,  $\tau_k = \max \{ \xi_1, \dots, \xi_{d_k} \}$  имеет фазовое распределение с параметрами  $(\alpha^{(k)}, \mathbf{A}^{(k)})$ , где

$$(\alpha^{(k)}, \mathbf{A}^{(k)}) = \bigvee_{i=1}^{d_k} (\hat{\alpha}^{(q_i)}, \hat{\mathbf{A}}^{(q_i)}).$$

Упростив (выделив одинаковые параметры фазовых распределений), приходим к утверждению теоремы.  $\square$

*Замечание 1.* Аналогичным образом может быть получено распределение длительности пребывания требований в сети массового обслуживания, которая не содержит дивайдеров и интеграторов. Поглощение в соответствующей модельной цепи Маркова отображает возвращение требования в источник требований. Начальное распределение определяется исходя из тех базовых систем, в которые требование может поступить непосредственно из источника.

*Замечание 2.* Элементарная подсеть  $\mathcal{H}_k$  может быть представлена как система типа  $\cdot / \text{PH}(\alpha^{(k)}, \mathbf{A}^{(k)}) / \infty$ .

Теорема 1 позволяет рассмотреть процесс обслуживания фрагмента в элементарной подсети как эволюцию некоторой поглощающей цепи Маркова. Данную цепь Маркова можно рассматривать также как модель процесса обслуживания фрагмента в подсети обслуживания, которая не содержит дивайдеров и интеграторов (то есть все системы обслуживания — базовые системы).

Рассмотрим процедуру модификации исходной сети обслуживания  $\mathcal{N}$ , при которой элементарная подсеть  $\mathcal{H}_k$  замещается на другую подсеть, не содержащую

дивайдеров и интеграторов, с таким же распределением длительности пребывания фрагментов в ней. Указанную процедуру будем называть *редукцией сети обслуживания*  $\mathcal{N}$  относительно элементарной подсети  $\mathcal{H}_k$ .

**Определение 7.** Пусть  $\text{PH}(\boldsymbol{\alpha}^{(k)}, \mathbf{A}^{(k)})$  есть распределение длительности реакции для элементарной подсети  $\mathcal{H}_k$  с  $Y$  фазами,

$$\boldsymbol{\alpha}^{(k)} = \left( \alpha_i^{(k)} \right),$$

$$\mathbf{A}^{(k)} = \left( a_{i,j}^{(k)} \right), \quad i, j = 1, \dots, Y.$$

Интенсивности перехода  $a_i^*$  модельной цепи Маркова данного фазового распределения из состояния  $i = 1, \dots, Y$ , в поглощающее состояние имеют вид

$$a_i^* = - \sum_{j=1}^Y a_{i,j}^{(k)}.$$

В результате редукции будет получена новая сеть обслуживания  $\tilde{\mathcal{N}}$ , построенная по следующему алгоритму.

Обозначение параметров сети  $\tilde{\mathcal{N}}$  отличается от аналогичных параметров сети  $\mathcal{N}$  наличием знака  $\sim$  над обозначением параметра.

---

Алгоритм редукции сети обслуживания относительно подсети  $\mathcal{H}_k$

---

1. Положить параметры сети  $\tilde{\mathcal{N}}$  равными параметрам сети  $\mathcal{N}$ .
2. Определить множество новых базовых систем  $\mathcal{S}' = \{S'_1, \dots, S'_Y\}$ . Интенсивность обслуживания  $\mu'_i$  базовой системы  $S'_i$ ,  $i = 1, \dots, Y$ , полагается

$$\mu'_i := -a_{i,i}^{(k)}. \quad (21)$$

3. Добавить базовые системы из множества  $\mathcal{S}'$  в множество базовых систем сети  $\tilde{\mathcal{N}}$ . В каждую из матриц  $\tilde{\Theta}^{(l)}$ ,  $l = 0, 1, \dots, K$ , ввести  $Y$  новых строк и столбцов, отображающих переходы фрагментов для базовых систем из множества  $\mathcal{S}'$ , таким образом, что элементы строки и столбца с номером  $L + i$  соответствуют вероятностям переходов для базовой системы  $S'_i$ ,  $i = 1, \dots, Y$ .

Определить элементы матрицы передачи следующим образом:

$$\tilde{\theta}_{L+i, L+j}^{(l)} := \frac{a_{i,j}^{(k)}}{\mu'_i}, \quad i \neq j, \quad (22)$$

$$\tilde{\theta}_{L+i, L+i}^{(l)} := 0, \quad (23)$$

$$i, j = 1, \dots, Y, \quad l = 0, \dots, K.$$

4. Для  $l = 0, \dots, K$ , положить:

$$\tilde{\theta}_{i,L+j}^{(l)} := \alpha_j^{(k)} \theta_{i,x_k^{\mathcal{F}}}^{(l)}, \quad i = 0, \dots, L, j = 1, \dots, Y, \quad (24)$$

$$\tilde{\theta}_{L+i,j}^{(l)} := \frac{a_i^*}{\mu_i} \theta_{x_k^{\mathcal{J}},j}^{(l)}, \quad i = 1, \dots, Y, j = 0, \dots, L. \quad (25)$$

5. Исключить матрицу  $\tilde{\Theta}^{(k)}$  из набора матриц передач  $\tilde{\Theta}$ .
6. Исключить из множества систем обслуживания сети  $\tilde{\mathcal{N}}$  дивайдер  $F_k$  и интегратор  $J_k$ , а также соответствующие им строки и столбцы из набора матриц передач  $\tilde{\Theta}$ .
7. Все базовые системы  $\{S_i : i \in X_S\}$  сети  $\mathcal{N}$ , в которые поступал поток только  $k$ -фрагментов, исключить из сети  $\tilde{\mathcal{N}}$ . Из набора матриц передач  $\tilde{\Theta}$  исключаются соответствующие строки и столбцы.

Конец алгоритма.

**Теорема 2.** *Длительность пребывания требований в сети массового обслуживания  $\mathcal{N}$  с делением и слиянием требований имеет фазовое распределение.*

*Доказательство.* Пусть для некоторого  $k \in \{1, \dots, K\}$ ,  $\mathcal{H}_k$  является элементарной подсетью в сети обслуживания  $\mathcal{N}$ . Выполним процедуру редукции относительно  $\mathcal{H}_k$ , в результате получим новую сеть массового обслуживания  $\tilde{\mathcal{N}}_1$ . Если в сети обслуживания  $\tilde{\mathcal{N}}_1$  будет существовать элементарная подсеть, то выполним редукцию снова относительно найденной элементарной подсети. Таким образом, получаем последовательность  $\tilde{\mathcal{N}}_1, \tilde{\mathcal{N}}_2, \dots, \tilde{\mathcal{N}}_K$  сетей массового обслуживания. Сеть  $\tilde{\mathcal{N}}_K$  не будет содержать дивайдеров и интеграторов, а распределение длительности пребывания требований в ней может быть получено исходя из следствия теоремы 1.  $\square$

### 3. Пример

Рассмотрим сеть массового обслуживания с делением и слиянием требований, топология которой представлена ориентированным графом (Рис. 1), вершины которого обозначают системы обслуживания, а дуги соответствуют переходам фрагментов между системами сети.

Сеть обслуживания состоит из  $L = 10$  систем,  $S = \{S_1, \dots, S_6\}$ ,  $\mathcal{F} = \{F_1, F_2\}$ ,  $\mathcal{J} = \{J_1, J_2\}$ ,  $N_0$  — источник требований.

Дуга из  $N_i$  в  $N_j$  имеет метку  $(\theta_{i,j}^{(0)}, \theta_{i,j}^{(1)}, \theta_{i,j}^{(2)})$ . Например, дуга с меткой  $(0; 0,5; 1)$ , ведущая из  $S_2$  в  $S_4$ , означает, что 1-фрагменты переходят из  $S_2$  в  $S_4$  с вероятностью 0,5, для 2-фрагментов этот же переход осуществляется с вероятностью 1. Вероятность поступления требования из источника  $N_0$  в  $F_1$  равна 0,5. Требование, поступившее в  $F_1$ , делится на два 1-фрагмента, один из которых поступает в  $S_1$ , а другой — в  $S_2$ .

Пусть  $\Lambda_0 = 1$ ,  $\mu_i = 2$ ,  $i = 1, \dots, 6$ .

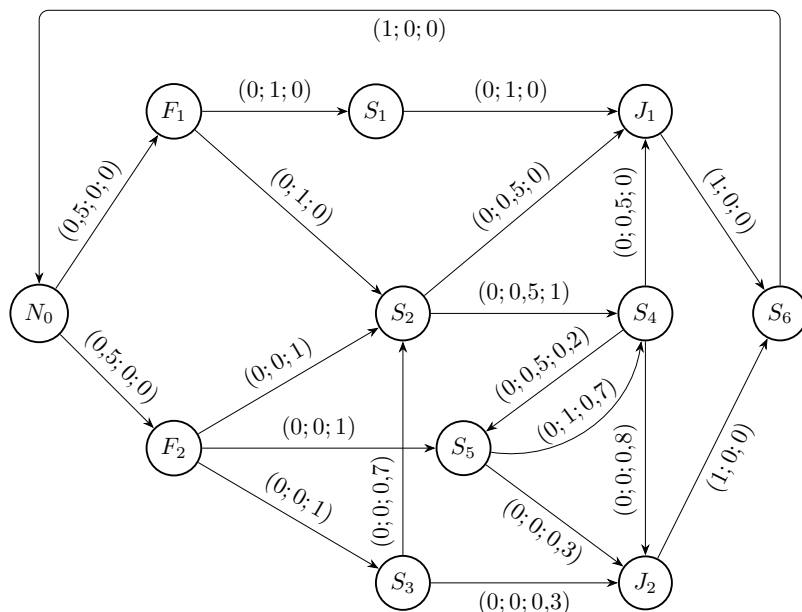


Рис. 1: Пример сети обслуживания с делением и слиянием требований

Суммарные интенсивности входящих потоков фрагментов для систем обслуживания сети:

$$\begin{aligned}
 \Lambda_{in}(S_1) &= 0,5, & \Lambda_{in}(S_2) &= 1,35, \\
 \Lambda_{in}(S_3) &= 0,5, & \Lambda_{in}(S_4) &= 1,8953, \\
 \Lambda_{in}(S_5) &= 1,0291, & \Lambda_{in}(S_6) &= 1, \\
 \Lambda_{in}(F_1) &= 0,5, & \Lambda_{in}(F_2) &= 0,5, \\
 \Lambda_{in}(J_1) &= 1, & \Lambda_{in}(J_2) &= 1,5.
 \end{aligned}$$

Рассмотрим распределение длительности реакции  $\tau_1$  для элементарной подсети  $\mathcal{H}_1 = \{F_1, S_1, S_2, S_4, S_5, J_1\}$ .

Из теоремы 1 получаем:

$$\mathcal{A}(1) = \{1, 2\},$$

$$\mathcal{B}(1, S_1) = (1), \quad \mathcal{B}(1, S_2) = (2, 4, 5);$$

параметры фазового распределения

$$\hat{\alpha}^{(1)} = (1), \quad \hat{A}^{(1)} = (-2);$$

$$\hat{\alpha}^{(2)} = (1, 0, 0), \quad \hat{A}^{(2)} = \begin{pmatrix} -2 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 2 & -2 \end{pmatrix};$$

тогда параметры распределения длительности реакции подсети  $\mathcal{H}_1$

$$\alpha^{(1)} = (1, 0, 0, 0, 0, 0, 0),$$

$$\mathbf{A}^{(1)} = \begin{pmatrix} -4 & 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & -4 & 1 & 1 & 0 & 2 & 0 \\ 0 & 2 & -4 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix}.$$

После выполнения редукций получаем, что математическое ожидание  $E(\tau)$  длительности пребывания требований в сети обслуживания  $\mathcal{N}$  и дисперсия  $Var(\tau)$  длительности пребывания требований равны:

$$E(\tau) = 2,2027, \quad Var(\tau) = 1,8736.$$

### Заключение

В данной работе рассмотрено обобщение классических сетей массового обслуживания с делением и слиянием требований и получено распределение длительности пребывания требований в сетях данного класса. В качестве предмета дальнейших исследований может быть рассмотрен случай одноприборных базовых систем обслуживания.

### Список литературы

- [1] Thomasian A. Analysis of fork/join and related queueing systems // ACM Computing Surveys. 2014. Vol. 47, № 2. Pp. 1–71.
- [2] Rizk A., Poloczek F., Ciucu F. Computable bounds in fork-join queueing systems // ACM SIGMETRICS Performance Evaluation Review. 2015. Vol. 43, № 1. Pp. 335–346. <http://dx.doi.org/10.1145/2796314.2745859>
- [3] Joshi G., Liu Y., Soljanin E. On the delay-storage trade-off in content download from coded distributed storage systems // IEEE Journal on Selected Areas in Communications. 2014. Vol. 32, № 5. Pp. 989–997.
- [4] Горбунова А.В., Зарядов И.С., Матюшенко С.И., Самуйлов К.Е., Шоргин С.Я. Аппроксимация времени отклика системы облачных вычислений // Информатика и ее применения. 2015. Т. 9, № 3. С. 31–38.
- [5] Heidelberger P., Trivedi K.S. Analytic queueing models for programs with internal concurrency // IEEE Transactions on Computers. 1983. Vol. 32, № 1. Pp. 73–82. <https://doi.org/10.1109/TC.1983.1676125>
- [6] Flatto L., Hahn S. Two parallel queues created by arrivals with two demands I // SIAM Journal on Applied Mathematics. 1984. Vol. 44, № 5. Pp. 1041–1053. <https://doi.org/10.1137/0144074>
- [7] Nelson R., Tantawi A.N. Approximate analysis of fork/join synchronization in parallel queues // IEEE Transactions on Computers. 1988. Vol. 37, № 6. Pp. 739–743. <https://doi.org/10.1109/12.2213>

- [8] Ko S.-S., Serfozo R.F. Response times in M/M/s fork-join networks // *Advances in Applied Probability*. 2004. Vol. 36, № 3. Pp. 854–871. <https://doi.org/10.1239/aap/1093962238>
- [9] Ko S.-S., Serfozo R.F. Sojourn times in G/M/1 fork-join networks // *Naval Research Logistics*. 2008. Vol. 55, № 5. Pp. 432–443.
- [10] Baccelli F., Makowski A.M., Shwartz A. The fork-join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds // *Advances in Applied Probability*. 1989. Vol. 21, № 3, Pp. 629–660. <https://doi.org/10.1017/S0001867800018851>
- [11] Latouche G., Ramaswami V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial & Applied Mathematics (SIAM), 1999.
- [12] He Q.-M. *Fundamentals of Matrix-Analytic Methods*. New York, USA: Springer, 2014. 349 p.
- [13] Neuts M.F. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Baltimore: The Johns Hopkins University Press, 1981.
- [14] Lipsky L. *Queueing Theory: A Linear Algebraic Approach*. New York, USA: Springer, 2009. 548 p.
- [15] Башарин Г.П., Бочаров П.П., Коган Я.А. *Анализ очередей в вычислительных сетях. Теория и методы расчета*. М.: Наука, 1989. 336 с.

#### Библиографическая ссылка

Осипов О.А., Тананко И.Е. Сети массового обслуживания произвольной топологии с делением и слиянием требований: случай бесконечноприборных систем обслуживания // *Вестник ТвГУ. Серия: Прикладная математика*. 2017. № 4. С. 43–58. <https://doi.org/10.26456/vtprm188>

#### Сведения об авторах

1. **Осипов Олег Александрович**

аспирант Саратовского государственного университета имени Н.Г. Чернышевского.

*Россия, 410012, г. Саратов, ул. Астраханская, 83, СГУ имени Н.Г. Чернышевского. E-mail: [oleg.alex.osipov@gmail.com](mailto:oleg.alex.osipov@gmail.com).*

2. **Тананко Игорь Евстафьевич**

заведующий кафедрой системного анализа и автоматического управления Саратовского государственного университета имени Н.Г. Чернышевского.

*Россия, 410012, г. Саратов, ул. Астраханская, 83, СГУ имени Н.Г. Чернышевского. E-mail: [tanankoie@info.sgu.ru](mailto:tanankoie@info.sgu.ru).*

## FORK-JOIN QUEUEING NETWORKS WITH AN ARBITRARY TOPOLOGY: THE INFINITE SERVER CASE

**Osipov Oleg Aleksandrovich**

PhD student, Saratov State University  
Russia, 410012, Saratov, 83 Astrahanskaya Str., SSU.  
E-mail: [oleg.alex.osipov@gmail.com](mailto:oleg.alex.osipov@gmail.com)

**Tananko Igor Evstafevich**

Head of System Analysis and Automatic Control department,  
Saratov State University  
Russia, 410012, Saratov, 83 Astrahanskaya Str., SSU.  
E-mail: [tanankoie@info.sgu.ru](mailto:tanankoie@info.sgu.ru)

---

Received 29.06.2017, revised 11.12.2017.

---

This article introduces a generalization of the classical parallel-server fork-join queueing networks. We consider open fork-join queueing networks with an arbitrary topology. There are three types of nodes in the queueing networks (service-nodes, fork-nodes, join-nodes). We focus on the infinite server case, in which each service-node is a queueing system with infinite servers. An arriving job is split (at fork-nodes) into a number of independent tasks that are serviced in parallel (at multiple service-nodes). Each task can be split at fork-nodes repetitively. These tasks synchronize (at join-nodes) before they leave the network. A method, which allows to obtain the sojourn time distribution, was developed.

**Keywords:** fork-join networks, queueing network, sojourn time distribution, performance evaluation, synchronization, parallel processing, PH-distribution.

### Bibliographic citation

Osipov O.A., Tananko I.E. Fork-join queueing networks with an arbitrary topology: the infinite server case. *Vestnik TsvGU. Seriya: Prikladnaya Matematika* [Herald of Tver State University. Series: Applied Mathematics], 2017, no. 4, pp. 43–58. (in Russian) <https://doi.org/10.26456/vtpmk188>

### References

- [1] Thomasian A. Analysis of fork/join and related queueing systems. *ACM Computing Surveys*, 2014, vol. 47(2), pp. 1–71.
- [2] Rizk A., Poloczek F., Ciucu F. Computable bounds in fork-join queueing systems. *ACM SIGMETRICS Performance Evaluation Review*, 2015, vol. 43(1), pp. 335–346. <http://dx.doi.org/10.1145/2796314.2745859>

- [3] Joshi G., Liu Y., Soljanin E. On the delay-storage trade-off in content download from coded distributed storage systems. *IEEE Journal on Selected Areas in Communications*, 2014, vol. 32(5), pp. 989–997.
- [4] Gorbunova A.V., Zaryadov K.S., Matyushenko S.I., Samuylov K.E., Shorgin S.Ya. The approximation of response time of a cloud computing system. *Informatika i ee Primeneniya* [Informatics and Applications], 2015, vol. 9(3), pp. 31–38. (in Russian)
- [5] Heidelberger P., Trivedi K.S. Analytic queueing models for programs with internal concurrency. *IEEE Transactions on Computers*, 1983, vol. 37(6), pp. 73–82. <https://doi.org/10.1109/TC.1983.1676125>
- [6] Flatto L., Hahn S. Two parallel queues created by arrivals with two demands I. *SIAM Journal on Applied Mathematics*, 1984, vol. 44(5), pp. 1041–1053. <https://doi.org/10.1137/0144074>
- [7] Nelson R., Tantawi A.N. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Transactions on Computers*, 1988, vol. 37(6), pp. 739–743. <https://doi.org/10.1109/12.2213>
- [8] Ko S.-S., Serfozo R.F. Response times in M/M/s fork-join networks. *Advances in Applied Probability*, 2004, vol. 36(3), pp. 854–871. <https://doi.org/10.1239/aap/1093962238>
- [9] Ko S.-S., Serfozo R.F. Sojourn times in G/M/1 fork-join networks. *Naval Research Logistics*, 2008, vol. 55(5), pp. 432–443.
- [10] Baccelli F., Makowski A.M., Shwartz A. The fork-join queue and related systems with synchronization constraints: Stochastic ordering and computable bounds. *Advances in Applied Probability*, 1989, vol. 21(3), pp. 629–660. <https://doi.org/10.1017/S0001867800018851>
- [11] Latouche G., Ramaswami V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial & Applied Mathematics (SIAM), 1999.
- [12] He Q.-M. *Fundamentals of Matrix-Analytic Methods*. Springer, New York, 2014. 349 p.
- [13] Neuts M.F. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, Baltimore, 1981.
- [14] Lipsky L. *Queueing Theory: A Linear Algebraic Approach*. Springer, New York, 2009. 548 p.
- [15] Basharin G.P., Bocharov P.P., Kogan Ya.A. *Analiz Ocheredey v Vychislitel'nykh Setyakh. Teoriya i Metody Rascheta* [Analysis of Queues in Computer Networks. Theory and Computational Methods]. Nauka Publ., Moscow, 1989. 336 p. (in Russian)