

## ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ СЕТЕВЫХ РЕСУРСОВ В СИСТЕМАХ С СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРОЙ

### Введение

Концепция сервис-ориентированной архитектуры (service-oriented architecture, SOA) предлагает эффективный подход к решению одной из самых сложных и насущных проблем современных информационных систем – проблемы интеграции приложений. Идея SOA заключается в создании архитектурной платформы, в которой все функции определены как независимые распределенные компоненты – сервисы. Взаимодействуя по сети в определенной последовательности, сервисы позволяют реализовать тот или иной бизнес-процесс. В качестве сервиса в SOA может выступать как целое приложение, так и отдельные его функциональные модули. Представление компонентов как сервисов со стандартными интерфейсами обеспечивает их многократное использование для формирования новых приложений и расширения возможностей уже существующих.

Однако одним из существенных недостатков информационных систем на основе SOA является снижение производительности, возникающее за счет использования сети. В связи с этим актуальной задачей является разработка механизмов координации работы распределенных системных компонентов с учетом структурных и функциональных свойств базовой телекоммуникационной сети с целью повышения производительности SOA-системы в целом.

### Принципы построения систем SOA

Сервис-ориентированная архитектура основана на следующих базовых принципах [1]:

1. Слабая связанность (Loosely coupled). С точки зрения реализации, сервисы в SOA независимы друг от друга: они выполняют определенные действия по запросам, полученным от других сервисов, и возвращают результаты. Все детали этого выполнения полностью скрыты: в концепции SOA сервисы – это «черные ящики». Отсюда следует, что изменения в реализации сервиса никак не повлияют на прикладной компонент, который этот сервис использует, и наоборот. Слабая связанность обеспечивает простую и быструю адаптацию системы к изменениям в структуре и принципах реализации сервисов.

2. «Крупнозернистая» (coarse-grained) структура сервисов. Сервисы в SOA представляют собой модули бизнес-логики достаточно высокого уровня, благодаря чему взаимодействие между ними сводится к ограниченному числу сообщений по содержанию бизнес-логики вместо множества низкоуровневых вызовов, учитывающих детали реализации сервисов. Такой подход снижает нагрузку на сеть и способствует более высокой производительности системы. Важно, что с точки зрения архитектуры сервис, независимо от внутренней структуры и языка реализации, выглядит как единое целое.

3. Стандартные интерфейсы (standards-based). В SOA вместо частных, закрытых API, предлагаемых тем или иным поставщиком, используются интерфейсы, нейтральные по отношению к специфике реализации сервиса, которая накладывается в результате учета особенностей аппаратной платформы, операционной системы, языка программирования. Подобный нейтралитет обеспечивает универсальность взаимодействия сервисов в разнородной среде и снижает затраты на интеграцию.

В рамках идеологии SOA интеграционные механизмы не требуют знания деталей реализации бизнес-функций, которые предоставляет сервис, и способов передачи и получения информации от одного сервиса к другому, но обеспечивают средства для декларации возможностей сервиса всем компонентам по сети, которые могут быть заинтересованы в их использовании.

Взаимодействие сервисов осуществляется по принципу «публикация-поиск-соединение» (рис.1): приложение, предоставляющее определенный сервис (провайдер сервиса), размещает информацию о нем в каталоге сервисов. Потребитель сервиса – приложение, которому необходима функциональность данного сервиса, находит информацию о нем в каталоге, для того чтобы установить соединение с этим сервисом и послать запрос.

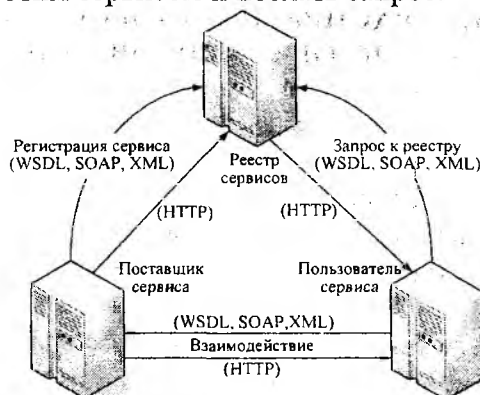


Рис. 1

На данном этапе своего развития сервис-ориентированные архитектуры для описания и организации взаимодействия используют базовые стандарты Web-сервисов:

- eXtensible Markup Language (XML) – для представления данных;
- Web Services Definition Language (WSDL) – для описания доступных Web-сервисов;
- Universal Description, Discovery, Integration (UDDI) – для создания каталога доступных по сети Web-сервисов;
- Simple Object Access Protocol (SOAP) – для обмена данными.

### **Постановка задачи оптимизации распределения сетевых ресурсов в системах с сервис-ориентированной архитектурой**

Реализация SOA связана с внедрением комплекса технологических решений, одним из которых является инфраструктура интеграции и управления сервисами. Для организации взаимодействия сервисов необходима среда, которая обеспечит динамическую маршрутизацию запросов от прикладного компонента – потребителя сервиса и получение результатов от приложения – провайдера сервиса. Для этого может потребоваться поддержка синхронных и асинхронных коммуникаций более низкого уровня между приложениями, трансформация и высокоскоростное распределение данных, трансляция протоколов, кэширование функций Web-сервисов, виртуализация ввода/вывода и т. д. Для решения этих задач все большее распространение получает технология корпоративной сервисной шины (Enterprise Service Bus, ESB), которая предоставляет единый механизм для передачи запросов и получения результатов сервисов, выполнения необходимых преобразований сообщений и транспортных протоколов и управления потоком обращений к сервисам. Благодаря такому управлению выполняется необходимая последовательность вызовов сервиса для реализации бизнес-процесса.

Провайдер обычно предоставляет сервисы, которые в свою очередь реализуют какой-либо сценарий обращений к вспомогательным сервисам. Тем самым в процессе предоставления сервиса конечному пользователю возникает взаимодействие между несколькими поставщиками, как показано на рис.2, где каждое взаимодействие между парой поставщиков наборов сервисов X, Y и Z (или поставщиком и конечным пользователем) охарактеризовано тремя параметрами:  $\lambda^{(k)}$  – интенсивность поступления запросов на сервис типа  $k$ ,  $V^{(k)}$  – объем данных, передаваемых по телекоммуникационной сети между серверами поставщиков при обслуживании запроса на сервис типа  $k$ ,  $B^{(k)}$  – количество запросов к серверу при обслуживании запроса на сервис типа  $k$ .

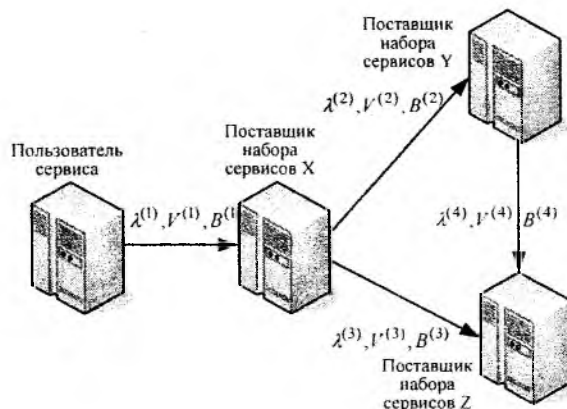


Рис. 2

На практике схема взаимодействия между поставщиками значительно усложняется вследствие того, что однотипные сервисы могут предоставляться различными поставщиками, конкурирующими друг с другом. В этом случае возникает задача оптимального выбора поставщика сервиса и распределения сетевых ресурсов для обслуживания потоков запросов на сервисы, возникающих между серверами различных поставщиков. На рис. 3 представлен граф, описывающий взаимодействие между пользователями и поставщиками в процессе предоставления сервиса, сценарий которого отображен на рис. 2, в случае, когда в SOA-системе для предоставления сервиса каждого типа существует два конкурирующих поставщика, а доступ пользователей к сервисам обеспечивают пять узлов агрегирования доступа.

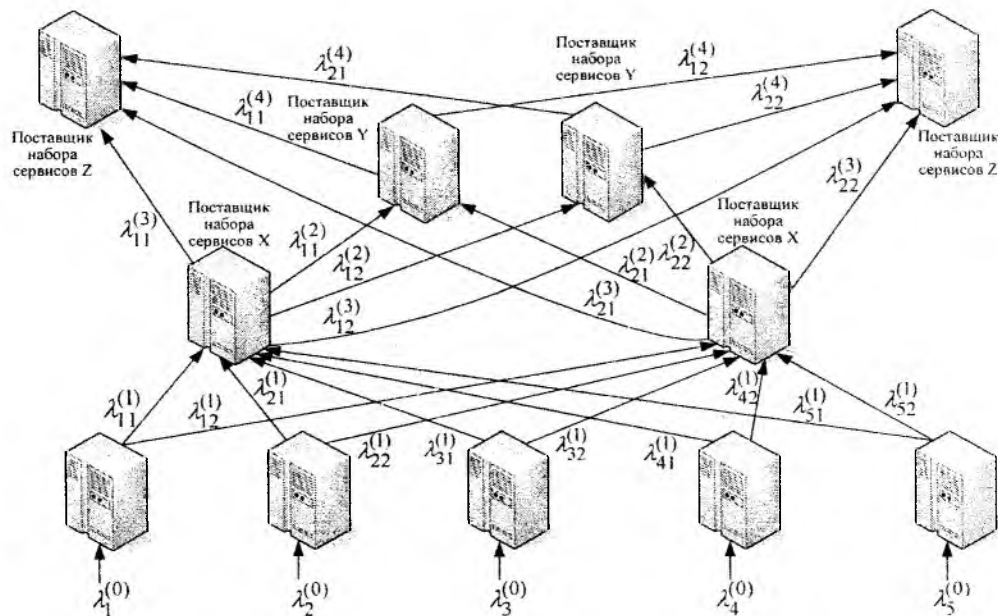


Рис. 3

Для потоков запросов между распределенными компонентами SOA-системы введено обозначение  $\lambda_{ij}^{(k)}$ , где  $k$  – тип услуги,  $i$  и  $j$  – номера серверов, запрашивающего и предоставляющего услугу типа  $k$  соответственно. Необходимо отметить, что в общем случае для произвольного сервера  $m$   $\sum_{i,k} \lambda_{im}^{(k)} \neq \sum_{i,k} \lambda_{mi}^{(k)}$ . Для учета данного факта введем весовые коэффициенты  $A_{st}$ , показывающие количество запросов сервиса типа  $t$ , возникающее при посту-

плении запроса сервиса типа  $s$ . Для сервиса, сценарий которого отображен на рис. 2, необходимо определить весовые коэффициенты  $A_{12}$ ,  $A_{13}$ ,  $A_{24}$ .

Задачу оптимального распределения сетевых ресурсов в рассматриваемой системе SOA можно сформулировать следующим образом: определить оптимальные значения интенсивностей потоков запросов  $\lambda_{ij}^{(k)}$  между распределенными компонентами SOA-системы, при котором будет обслужено максимальное количество запросов сервиса от конечных пользователей, а сетевые ресурсы системы (каналы связи телекоммуникационной сети и серверы) не будут перегружены.

Математическая постановка данной задачи может быть представлена как задача линейного программирования в следующем виде: максимизировать

$$f(\lambda) = \lambda_1^{(0)} + \lambda_2^{(0)} + \lambda_3^{(0)} + \lambda_4^{(0)} + \lambda_5^{(0)} = \sum_{i=1}^5 \sum_{j=1}^2 \lambda_{ij}^{(1)} \rightarrow \max \quad (1)$$

при следующих ограничениях:

$$\left\{ \begin{array}{l} A_{12} \sum_{i=1}^5 \lambda_{im}^{(1)} = \sum_{j=1}^2 \lambda_{mj}^{(2)}, m = 1,2 \\ A_{13} \sum_{i=1}^5 \lambda_{im}^{(1)} = \sum_{j=1}^2 \lambda_{mj}^{(3)}, m = 1,2 \\ A_{24} \sum_{i=1}^2 \lambda_{im}^{(2)} = \sum_{j=1}^2 \lambda_{mj}^{(4)}, m = 1,2 \\ B^{(1)} \sum_{i=1}^5 \lambda_{im}^{(1)} \leq \alpha_m^{(X)} \mu_m^{(X)}, m = 1,2 \\ B^{(2)} \sum_{i=1}^2 \lambda_{im}^{(2)} \leq \alpha_m^{(Y)} \mu_m^{(Y)}, m = 1,2 \\ B^{(3)} \sum_{i=1}^2 \lambda_{im}^{(3)} + B^{(4)} \sum_{i=1}^2 \lambda_{im}^{(4)} \leq \alpha_m^{(Z)} \mu_m^{(Z)}, m = 1,2 \\ V^{(1)} \lambda_{ij}^{(1)} \leq \rho_{ij}^{(1)} C_{ij}^{(1)}, i = \overline{1,5}, j = 1,2 \\ V^{(2)} \lambda_{ij}^{(2)} \leq \rho_{ij}^{(2)} C_{ij}^{(2)}, i = 1,2, j = 1,2 \\ V^{(3)} \lambda_{ij}^{(3)} \leq \rho_{ij}^{(3)} C_{ij}^{(3)}, i = 1,2, j = 1,2 \\ V^{(4)} \lambda_{ij}^{(4)} \leq \rho_{ij}^{(4)} C_{ij}^{(4)}, i = 1,2, j = 1,2 \\ \lambda_{ij}^{(1)} \geq 0, i = \overline{1,5}, j = 1,2 \\ \lambda_{ij}^{(k)} \geq 0, i = 1,2, j = 1,2, k = \overline{2,4} \end{array} \right. \quad (2)$$

где  $\mu_m^{(X)}$ ,  $\mu_m^{(Y)}$ ,  $\mu_m^{(Z)}$  – производительности  $m$ -го сервера поставщика набора сервисов X, Y и Z соответственно;  $\alpha_m^{(X,Y,Z)}$  – предельное значение коэффициента загрузки  $m$ -го сервера поставщика набора сервисов X, Y и Z соответственно;  $C_{ij}^{(k)}$  – пропускные способности телекоммуникационных подсетей, связывающих серверы поставщиков  $i$  и  $j$  при предоставлении

сервиса типа  $k$ ;  $\rho_{ij}^{(k)}$  – предельные значения коэффициентов загрузки телекоммуникационных подсетей, связывающих серверы поставщиков  $i$  и  $j$  при предоставлении сервиса типа  $k$ .

### Пример решения задачи оптимального распределения сетевых ресурсов в системе с сервис-ориентированной архитектурой

Рассмотрим систему с сервис-ориентированной архитектурой, структура которой имеет вид, представленный на рис. 3. Исходные данные, необходимые для решения задачи оптимального распределения сетевых ресурсов в данной системе, приведены в табл. 1.

Таблица 1

Тип сервиса $k$	1	2	3	4
$B^{(k)}$	2,5	1	1,5	2
$V^{(k)}$ , кбайт	200	150	80	70

Значения производительностей серверов ( $\mu_m^{(X)}$ ,  $\mu_m^{(Y)}$ ,  $\mu_m^{(Z)}$ , *запрос/мс*) конкурирующих поставщиков наборов сервисов X, Y и Z приведены в табл. 2.

Таблица 2

Набор сервисов Поставщик набора сервисов	X	Y	Z
1	2,5	1	3,5
2	2	1,5	2,5

Пропускные способности ( $C_{ij}^{(k)}$ , *Мбит/с*) телекоммуникационных подсетей, связывающих серверы поставщиков  $i$  и  $j$  при предоставлении сервиса типа  $k$  приведены в табл. 3.

Таблица 3

$k$	$i \backslash j$		1	2	3	4	5
	1	2					
1	1		300	200	250	350	450
	2		650	750	500	700	600
2	1		300	500	–	–	–
	2		600	400	–	–	–
3	1		300	300	–	–	–
	2		300	300	–	–	–
4	1		300	300	–	–	–
	2		300	300	–	–	–

Задача оптимального распределения сетевых ресурсов в данном случае является задачей линейного программирования, математическая постановка которой приведена в виде целевой функции (3) или (4) и системы ограничений (5). Предельные значения всех коэффициентов загрузки были приняты одинаковыми  $\alpha = \rho = 0,5$

$$f(\lambda) = \lambda_{11}^{(1)} + \lambda_{12}^{(1)} + \lambda_{21}^{(1)} + \lambda_{22}^{(1)} + \lambda_{31}^{(1)} + \lambda_{32}^{(1)} + \lambda_{41}^{(1)} + \lambda_{42}^{(1)} + \lambda_{51}^{(1)} + \lambda_{52}^{(1)} \rightarrow \max; \quad (3)$$

$$f^*(\lambda) = -\lambda_{11}^{(1)} - \lambda_{12}^{(1)} - \lambda_{21}^{(1)} - \lambda_{22}^{(1)} - \lambda_{31}^{(1)} - \lambda_{32}^{(1)} - \lambda_{41}^{(1)} - \lambda_{42}^{(1)} - \lambda_{51}^{(1)} - \lambda_{52}^{(1)} \rightarrow \min; \quad (4)$$

$$\begin{cases}
A_{12}(\lambda_{11}^{(1)} + \lambda_{21}^{(1)} + \lambda_{31}^{(1)} + \lambda_{41}^{(1)} + \lambda_{51}^{(1)}) = \lambda_{11}^{(2)} + \lambda_{12}^{(2)} \\
A_{12}(\lambda_{12}^{(1)} + \lambda_{22}^{(1)} + \lambda_{32}^{(1)} + \lambda_{42}^{(1)} + \lambda_{52}^{(1)}) = \lambda_{21}^{(2)} + \lambda_{22}^{(2)} \\
A_{13}(\lambda_{11}^{(1)} + \lambda_{21}^{(1)} + \lambda_{31}^{(1)} + \lambda_{41}^{(1)} + \lambda_{51}^{(1)}) = \lambda_{11}^{(3)} + \lambda_{12}^{(3)} \\
A_{13}(\lambda_{12}^{(1)} + \lambda_{22}^{(1)} + \lambda_{32}^{(1)} + \lambda_{42}^{(1)} + \lambda_{52}^{(1)}) = \lambda_{21}^{(3)} + \lambda_{22}^{(3)} \\
A_{24}(\lambda_{11}^{(2)} + \lambda_{21}^{(2)}) = \lambda_{11}^{(4)} + \lambda_{12}^{(4)} \\
A_{24}(\lambda_{12}^{(2)} + \lambda_{22}^{(2)}) = \lambda_{21}^{(4)} + \lambda_{22}^{(4)} \\
B^{(1)}(\lambda_{11}^{(1)} + \lambda_{21}^{(1)} + \lambda_{31}^{(1)} + \lambda_{41}^{(1)} + \lambda_{51}^{(1)}) \leq \alpha\mu_1^{(X)} \\
B^{(1)}(\lambda_{12}^{(2)} + \lambda_{22}^{(2)} + \lambda_{32}^{(1)} + \lambda_{42}^{(1)} + \lambda_{52}^{(1)}) \leq \alpha\mu_2^{(X)} \\
B^{(2)}(\lambda_{11}^{(2)} + \lambda_{21}^{(2)}) \leq \alpha\mu_1^{(Y)} \\
B^{(2)}(\lambda_{12}^{(1)} + \lambda_{22}^{(1)}) \leq \alpha\mu_2^{(Y)} \\
B^{(3)}(\lambda_{11}^{(3)} + \lambda_{21}^{(3)}) + B^{(4)}(\lambda_{11}^{(4)} + \lambda_{21}^{(4)}) \leq \alpha\mu_1^{(Z)} \\
B^{(3)}(\lambda_{12}^{(3)} + \lambda_{22}^{(3)}) + B^{(4)}(\lambda_{12}^{(4)} + \lambda_{22}^{(4)}) \leq \alpha\mu_2^{(Z)} \\
V^{(1)}\lambda_{ij}^{(1)} \leq \rho C_{ij}^{(1)}, i = \overline{1,5}, j = 1,2 \\
V^{(2)}\lambda_{ij}^{(2)} \leq \rho C_{ij}^{(2)}, i = 1,2, j = 1,2 \\
V^{(3)}\lambda_{ij}^{(3)} \leq \rho C_{ij}^{(3)}, i = 1,2, j = 1,2 \\
V^{(4)}\lambda_{ij}^{(4)} \leq \rho C_{ij}^{(4)}, i = 1,2, j = 1,2 \\
\lambda_{ij}^{(1)} \geq 0, i = \overline{1,5}, j = 1,2 \\
\lambda_{ij}^{(k)} \geq 0, i = 1,2, j = 1,2, k = \overline{2,4}
\end{cases} \quad (5)$$

Решение задачи было получено с помощью инструментария «Optimization Toolbox» программного пакета MatLab. Данное решение представлено на рис. 4 в виде полученных значений интенсивностей потоков между распределенными компонентами заданной SOA-системы. Полученное максимальное значение интенсивности поступления запросов сервиса от конечных пользователей рассмотренной SOA-системы, при котором ее сетевые ресурсы не будут перегружены, составило 375 запросов в секунду. На рис. 4 показаны полученные коэффициенты загрузки телекоммуникационных подсетей при оптимальном распределении сетевых ресурсов рассмотренной SOA-системы. В табл. 4 приведены рассчитанные коэффициенты загрузки всех шести серверов.

Таблица 4

Набор сервисов Поставщик набора сервисов	X	Y	Z
1	0,19	0,08	0,32
2	0,23	0,18	0,32

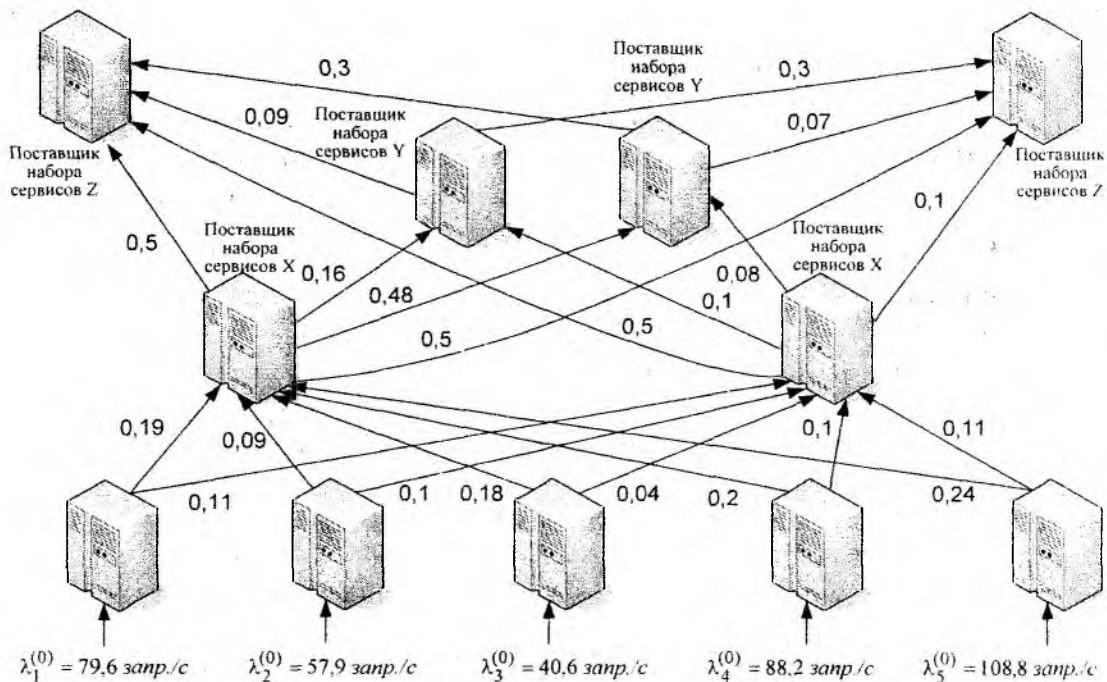


Рис. 4

## Выводы

Рассмотрены основные принципы построения информационных систем с сервис-ориентированной архитектурой. Поскольку данные системы являются по своей природе распределенными, их производительность наряду с другими факторами существенно зависит от характеристик телекоммуникационной сети, на базе которой организовано взаимодействие компонентов SOA-системы. В статье сформулирована задача распределения сетевых ресурсов системы SOA с целью максимизации ее производительности с учетом структурных и функциональных свойств базовой телекоммуникационной сети. Представлена математическая постановка данной задачи в виде задачи линейного программирования. Приведен пример решения данной задачи для SOA-системы с заданными структурными и функциональными характеристиками.

Предложенный метод оптимального распределения сетевых ресурсов применим для систем с сервис-ориентированной архитектурой в условиях отсутствия перегрузки. Направлением дальнейших исследований является разработка методов прогнозирования и предотвращения перегрузок в SOA-системах.

**Список литературы:** 1. Дубова Н. На пути к SOA // Директор ИС. 2005. №8. 2. Математичні основи теорії телекомунікаційних систем / За ред. В.В. Поповського. Харків: ТОВ «Компанія СМІТ», 2006. 564с. 3. Лысяк Т.Н. Распределение нагрузки в интеллектуальной сети // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2006. Вып.144. С.10–15. 4. Wellman M.P. A market oriented programming environment and its application to distributed multicommodity flow problems // Journal of Artificial Intelligence Research. 1993. №1. С. 1–23.

Харьковский национальный  
университет радиоэлектроники

Поступила в редколлегию 01.10.2009