

На правах рукописи

Бойко Павел Валентинович

**МАКС DSM:  
Система распределённой общей памяти  
для мультиагентных систем в IoT**

05.13.11 – Математическое и программное обеспечение вычислительных  
машин, комплексов и компьютерных сетей

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Санкт-Петербург – 2018

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего образования «Санкт-Петербургский государственный университет».

Научный руководитель: **Андрианов Сергей Николаевич**  
доктор физико-математических наук, профессор, заведующий кафедрой компьютерного моделирования и многопроцессорных систем федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет», почетный работник Высшей школы Российской Федерации

Официальные оппоненты: **Кореньков Владимир Васильевич**  
доктор технических наук, профессор, директор Лаборатории информационных технологий ОИЯИ, заведующий кафедрой распределенных информационно-вычислительных систем Международного университета «Дубна»

**Ковтуненко Алексей Сергеевич**  
кандидат технических наук, доцент кафедры информатики Федерального государственного бюджетного образовательного учреждения высшего образования «Уфимский государственный авиационный технический университет»

Ведущая организация: **Федеральное государственное бюджетное образовательное учреждение высшего образования «Петрозаводский государственный университет»**

Защита состоится 29 марта 2018 г. в 15:30 на заседании диссертационного совета Д 212.232.51 на базе Санкт-Петербургского государственного университета по адресу: 198504, Санкт-Петербург, Старый Петергоф, Университетский пр., 28, математико-механический факультет, ауд. 405.

С диссертацией можно ознакомиться в Научной библиотеке им. М. Горького Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., 7-9 и на сайте <https://disser.spbu.ru/files/disser2/disser/ryzpYcdgxc.pdf>.

Автореферат разослан «\_\_\_\_\_» \_\_\_\_\_ 2018 г.

Ученый секретарь диссертационного совета Д 212.232.51  
доктор физико-математических наук, профессор



Демьянович Юрий Казимирович

## Общая характеристика работы

**Актуальность темы исследования.** Мультиагентные системы (МАС) сформировались в виде отдельного научного направления в 1980-е, получили широкое признание в 1990-е и активно применяются на практике с 2000-х. Сегодня агентно-ориентированный подход используется для распределённого решения задач моделирования, организации работы коллективов роботов и др.

Ключевые темы исследований МАС – коммуникация, координация и согласование. При этом вопрос коммуникации является основополагающим, так как в отсутствие коммуникации другие взаимодействия в МАС становятся невозможны. Одной из основных концепций коммуникации в теории МАС является «доска объявлений» (англ. blackboard), обычно реализуемая посредством специализированного узла-сервера для хранения общей информации. В случае высокой динамики состава устройств в системе, характерной для сферы Интернета вещей (англ. Internet of Things – IoT), централизованный подход становится неэффективен и решением становится распределённая доска объявлений / распределённая общая память (англ. distributed shared memory – DSM).

Другим характерным свойством МАС в сфере IoT является применение энергоэффективных и относительно недорогих вычислителей, указанные свойства которых достигаются в том числе за счет снижения их производительности и отсутствия некоторых архитектурных блоков (например, блока управления памятью (англ. memory management unit – MMU)).

Указанные выше особенности построения МАС в сфере IoT привели к ситуации, когда базовый способ коммуникации, DSM, оказался в данной сфере недоступен. Среди причин имеются как технические (ориентация современных DSM систем на Linux-совместимые платформы с MMU), так и научные (модели и алгоритмы существующих DSM систем не учитывают специфику данной предметной области). В дополнение, рост интереса к IoT платформам и построению на них МАС – тенденция лишь последних лет, в связи с чем и научная и техническая составляющие для систем такого рода на данный момент существенно отстают от данных составляющих для традиционных систем.

**Степень разработанности темы исследования.** Масштабные работы по теории МАС были проведены в разные годы В. Б. Тарасовым, Y. Shoham, G. Weiss, M. Wooldridge и др. Многопроцессорные вычислительные системы

исследовались А. М. Андреевым, Г. П. Можаровым, В. В. Сюзевым и др.

Родоначальником DSM систем принято считать К. Li, защитившего диссертацию на данную тему в 1986 году. В дальнейшем множество ученых исследовали существующие и предлагали новые модели консистентности, балансируя между производительностью и предсказуемостью. Наиболее существенные результаты были получены такими учеными как N. Bershad, M. Dubois, K. Gharachorloo, J. R. Goodman, P. W. Hutto, P. Keleher, L. Lamport, R. J. Lipton и др. Разработкой и анализом алгоритмов занимались А. Forin, R. E. Kessler, O. Krieger, M. Livny, M. Stumm, S. Zhou и др. Конечные DSM системы создавали Н. Е. Bal, J. K. Bennett, В. N. Bershad, В. Fleisch, D. Gelernter, E. Jul, P. J. Keleher, К. Li, M. Stumm, L. Zeng, S. Zhou и др.

В последние годы направление исследований смещается от разработки принципиально новых моделей консистентности и универсальных алгоритмов к созданию узко-специализированных DSM-систем, наиболее полно учитывающих особенности той или иной предметной области (к примеру, можно обратиться к статьям L. Zeng 2017 года). Однако сфера IoT при высоком уровне востребованности оказалась до сих пор исследователями не охвачена.

**Объект и предмет исследования.** В данной работе в качестве объекта исследования выступают модели, методы, алгоритмы, языки и программные инструменты для организации процесса обмена «знаниями» в мультиагентных системах. Предметом исследования является мультиагентная координация посредством механизма распределённой общей памяти в условиях беспроводной связи в анизотропном радиоэфире.

**Цели и задачи диссертационной работы.** С учетом проведенного анализа, целью данной работы является разработка моделей, алгоритмов и программных средств, реализующих концепцию распределённой общей памяти для мультиагентных систем в IoT и позволяющих существенно упростить и ускорить создание прикладных решений в данной области. В соответствии с поставленной целью в работе решаются задачи по созданию следующих компонент.

1. Модели консистентности данных в распределённой системе, отвечающей требованиям и особенностям мультиагентных систем в области IoT.
2. Алгоритма организации узлов мультиагентной системы в само-восстанавливающуюся структуру, устойчивую к выходу из строя отдельных узлов.

3. Программного интерфейса для прикладного взаимодействия с разрабатываемым механизмом реализации концепции распределённой памяти.
4. Программного обеспечения, реализующего концепцию распределённой общей памяти для мультиагентных систем в сфере IoT.
5. Экспериментального программно-аппаратного стенда (включая сбор характеристик разработанного программного решения).

**Научная новизна** данного исследования заключается в следующем.

1. Усиленная модель консистентности по выходу (англ. enhanced release consistency) дополняет возможности известной ранее модели по выходу отдельными свойствами модели по входу. Данное сочетание свойств предложено впервые и позволяет добиться лучших характеристик в заданной предметной области, чем любая из исходных моделей.
2. Алгоритм ролей и переходов для узлов МАС, в отличие от описанных ранее алгоритмов, учитывает высокую динамичность системы и обеспечивает её устойчивость к сбоям отдельных узлов.
3. Концепция и интерфейс прикладного взаимодействия с DSM системой упрощают её использование и перенос на альтернативные аппаратные платформы, а также обеспечивают более высокий уровень защиты от ошибок прикладного программиста по сравнению с предложенными ранее.

**Теоретическая и практическая значимость.** Проведенное исследование стимулирует развитие МАС в сфере IoT. Разработанные методы и алгоритмы позволяют упростить создание прикладных решений в данной области. Результаты, изложенные в диссертации, могут быть использованы для разработки новых МАС систем, а также способов организации мультиагентного взаимодействия. При этом, впервые DSM система реализована для маломощных устройств без MMU, что расширяет возможности их применения.

Представленные результаты позволяют снизить «порог вхождения» в область создания ПО для МАС, предоставляя прикладным разработчикам простой способ координации множества устройств, снижая тем самым стоимость новых разработок в данной области.

Основные научные результаты диссертационной работы внедрены в коммерческий продукт ОСРВ МАКС (операционная система реального времени

для мультиагентных когерентных систем) и, вместе с ОС, используются в серийно производящемся оборудовании АО «ПКК Миландр»<sup>1</sup>.

**Методология и методы исследования.** Методология исследования характерна для области предметной инженерии и заключается в идентификации и анализе проблемы, формулировании цели и задач, анализа состояния исследований и существующей литературы по вопросу, проектировании решения, выборе средств и технологий, реализации, проведении экспериментов и апробации.

В качестве методов используются перечисленные ниже.

- эмпирический метод (анализ литературы);
- методы сравнения, обобщения, причинно-следственный (анализ существующих решений);
- метод индукции (формирование теоретического решения);
- методы объектно-ориентированного программирования (программная реализация);
- моделирование и эксперимент (анализ результатов реализации).

Кроме того, системный, причинно-следственный и сравнительный виды анализа были применены для получения всех основных научных результатов.

### **Положения, выносимые на защиту.**

1. Модель консистентности, позволяющая добиться лучших характеристик в заданной предметной области по сравнению с моделями, взятыми за основу.
2. Алгоритм ролей и переходов для узлов МАС, обеспечивающий устойчивость системы к сбоям отдельных узлов.
3. Концепция и интерфейс прикладного взаимодействия с DSM системой, упрощающие её использование и перенос на альтернативные аппаратные платформы, а также обеспечивающие более высокий уровень защиты от ошибок прикладного программиста.
4. Модель, алгоритм и концепция воплощены в программном решении МАКС DSM, произведены измерения характеристик решения на специально созданном оборудовании и программной имитационной модели.

---

<sup>1</sup>Один из ведущих российских разработчиков интегральных микросхем.

**Степень достоверности и апробация результатов.** Достоверность результатов работы обеспечивается анализом состояния исследований в данной области, докладами и публикациями по основным результатам, проведенными экспериментами и успешным внедрением.

Основные результаты диссертационного исследования докладывались и обсуждались на следующих мероприятиях.

- X Всероссийской межведомственной научной конференции «Актуальные направления развития систем охраны, специальной связи и информации для нужд органов государственной власти Российской Федерации» проводимой Академией Федеральной службы охраны Российской Федерации 7–8 февраля 2017 года в г.Орёл;
- IV Научно–практической конференции OS DAY «Операционная система как платформа», проводимой Институтом системного программирования РАН (ИСП РАН), 23–24 мая 2017 года в г.Москва;
- Семинаре «Актуальные проблемы создания бортовой системы навигации и навигационно-гидрографического обеспечения морских робототехнических комплексов (МРТК)», проводимом АО «ГНИНГИ» под руководством ФГБУ «ГНИИЦ РТ» 27 октября 2017 года в г.Санкт-Петербург.

Результаты исследования в виде программной реализации разработанных механизмов внедрены и являются существенной частью российской операционной системы реального времени МАКС.

Механизм распределённой общей памяти, интегрированный в ОСРВ МАКС, был внедрён и демонстрировался в работе на серийно выпускаемой АО «ПКК Миландр» продукции на 20-й Международной выставке электронных компонентов, модулей и комплектующих «ЭкспоЭлектроника» 25–27 апреля 2017 года.

**Публикации.** По основным теоретическим и практическим результатам диссертации лично автором опубликовано 5 статей [1–5] в журналах из перечня, рекомендованного ВАК Минобрнауки России.

Также автор является обладателем Свидетельства о государственной регистрации программы для ЭВМ № 2016617143 на ОСРВ МАКС (операционная система для мультиагентных когерентных систем) от 28 июня 2016 г., выданного Федеральной службой по интеллектуальной собственности [6].

**Личный вклад автора.** Все основные научные положения, выводы и рекомендации, составляющие содержание диссертационного исследования, получены автором лично.

**Структура и объем диссертации.** Диссертация состоит из введения, основной части (содержащей 3 главы), заключения, списка сокращений и условных обозначений, списка литературы, списка иллюстративного материала, списка таблиц и приложения. Общий объем диссертации – 133 стр., работа содержит 30 рис. и 4 табл. Список литературы включает 53 наименования на 7 страницах.

## **Основное содержание работы**

**Во введении** обоснована актуальность диссертационной работы, обозначены объект и предмет исследования, сформулирована цель и задачи, аргументирована научная новизна исследований, показана теоретическая и практическая значимость полученных результатов, обоснована достоверность и представлены выносимые на защиту научные положения.

**В первой главе** рассмотрены предпосылки и история развития концепции DSM, приведено её описание, проанализированы исследования, включая существующие модели консистентности, алгоритмы и программные реализации.

Развитие вычислительных систем привело к широкому распространению многопроцессорных архитектур. От архитектур с общей памятью, характерных для тесно связанных систем, наблюдается движение к слабосвязанным или распределённым системам, представляющим собой совокупность независимых вычислительных устройств, объединённых для решения некоторой задачи и формирующих таким образом мультиагентную систему, устройства которой общей памяти уже не имеют. Программирование распределённых систем оказывается гораздо более сложной задачей, чем разработка ПО для однопроцессорных или сильно связанных систем, что стимулирует поиски альтернативных концепций организации обмена информацией в подобных системах. Концепция DSM направлена на создание существенно более простого (с точки зрения прикладного программиста) способа организации коммуникаций в распределённых системах, скрывающего сложность реализации данной концепции под относительно простым интерфейсом, работа с которым напоминает программирование систем с общей памятью. По сравнению с моделью явной передачи данных, на смену

операциям Send и Receive (отправка и получение сообщений) выходят операции Read и Write (чтение и запись в/из памяти), что кардинально изменяет подход к программированию: на прикладном уровне DSM позволяет относительно легко создавать механизмы, традиционно считающиеся непростыми и характерными, ввиду высокой стоимости их разработки, скорее для «больших» систем (промышленных, военных, медицинских и др.)

Ранние реализации концепции DSM обладали низкой производительностью, что стимулировало активные исследования с целью дополнить удобство использования таких систем приемлемыми показателями производительности. Результатом стало создание множества моделей консистентности данных, наиболее поздние из которых позволяют достигать результатов производительности, сравнимых с результатами систем, созданных в рамках классической концепции обмена сообщениями. Однако так как условия функционирования систем и требования к ним со временем изменяются, продолжаются исследования, направленные на выработку новых моделей, более подходящих к тем или иным условиям использования.

С целью эффективной реализации той или иной модели консистентности множеством исследователей создаются алгоритмы, отвечающие разнообразным требованиям функционирования конкретных систем и обладающие различными характеристиками производительности и надёжности.

Параллельно различными учеными создаются конечные реализации DSM систем. При этом ранние реализации были скорее экспериментальными, демонстрирующими, например, возможность лёгкой адаптации программного обеспечения, разработанного для систем с общей памятью, к работе в распределённой среде. Постепенно решения становились всё более универсальными, а в последние годы тенденция вновь изменилась – сегодня DSM системы создаются для решения узконаправленных задач. При этом всё чаще DSM системы внедряются в состав более крупных систем, тем самым расширяя их возможности.

Несмотря на то, что концепция DSM применима к любым видам MAC или распределённых систем, сфера MAC в IoT – одна из наиболее бурно развивающихся, стимулирующая массовый интерес к распределённым системам – остаётся не охвачена данной концепцией. Сложившуюся ситуацию можно изменить, адаптировав DSM для данной сферы, что позволит заметно упростить создание

конечных распределённых решений и привнести в них сложную функциональность «больших систем», сократив при этом время на разработку. Параллельно возникают задачи адаптации или создания новых моделей консистентности и алгоритмов, более полно соответствующих требованиям сферы МАС для IoT.

**Во второй главе** уточнено назначение создаваемого в рамках данного исследования программного решения, получившего название МАКС DSM<sup>1</sup>, сформулированы требования (к решению и его окружению), предложена новая модель усиленной консистентности по выходу, разработан алгоритм ролей и переходов, а также описана концепция прикладного интерфейса (в виде требований к нему), учитывающая недостатки ранее созданных систем.

Разрабатываемое решение предназначено для использования в мультиагентных системах в сфере IoT, включая автоматизированные системы коммерческого учёта электроэнергии (АСКУЭ) и беспилотные летательные аппараты (БЛА, БПЛА). Создаваемая система призвана функционировать на маломощных микроконтроллерах без MMU, интегрируясь и задействуя в качестве нижележащего сетевого слоя операционную систему реального времени для мультиагентных когерентных систем (ОСРВ МАКС). Решение должно обеспечивать создание сетей из полутора десятков устройств, допускающих выход из строя отдельных узлов без потери общей функциональности.

Созданные ранее модели консистентности предлагают широкий выбор решений по критериям величины накладных сетевых расходов и удобства программирования. Так как одним из требований сферы IoT является отказоустойчивость, основной стратегией был избран подход широковещательных операций записи (англ. write-broadcast). Несмотря на то, что ранее данный подход не был популярен в связи с создаваемой им повышенной нагрузкой на сеть, в условиях радиоэффира (или иной разделяемой и принципиально широковещательной среды) его недостатки нивелируются, что позволяет эффективно задействовать его с целью минимизации времени, в течение которого отдельный узел системы может обладать уникальными данными (которые не могут быть восстановлены в случае выхода данного узла из строя). На основе модели консистентности по выходу, а также модели по входу, была создана новая модель, расширившая первую модель такими новыми для неё свойствами как наличие связи между

---

<sup>1</sup>По имени операционной системы реального времени МАКС, интеграция с которой была осуществлена.

общими и синхронизационными переменными и разделение захватов на эксклюзивные и неэксклюзивные. Данное сочетание свойств позволило добиться лучших характеристик в заданной предметной области чем любая из исходных моделей.

Описание модели состоит из шести правил и приведено в разделе 2.2.1. диссертации. Схематично данную модель можно выразить рисунком 1.

$$\begin{array}{l}
 P_1: \quad S(X) \quad W(x)1 \quad W(x)2 \quad F(X) \\
 \hline
 P_2: \quad \quad \quad \quad \quad \quad \quad S(X) \quad R(x)2 \quad F(X) \\
 \hline
 P_3: \quad R(x)1
 \end{array}$$

Рисунок 1 – Модель расширенной консистентности по выходу

В данной нотации процессы системы расположены по вертикали, а шкала времени по горизонтали.  $P_N$  – процесс номер  $N$ . Каждый процесс выполняется на отдельном узле.  $R$  и  $W$  – операции чтения и записи (от англ. read и write). Тройка  $op(var)val$  означает операцию над памятью, где  $op$  – вид операции ( $R$  или  $W$ ),  $var$  – обозначение адресуемой переменной, а  $val$  – значение, являющееся входным для операции записи или же выходным в случае операции чтения.  $S(X)$  – вход в критическую секцию  $X$  (от англ. start),  $F(X)$  – выход из неё (от англ. finish). Подразумевается, что синхронизационная переменная  $X$  проассоциирована с общей переменной  $x$ . Изначально все переменные равны нулю.

Как и в модели консистентности по входу, консистентность в предложенной модели обеспечивается исключительно после входа в соответствующую критическую секцию (что можно видеть на примере процессов  $P_2$  и  $P_3$ ). Однако операции входа в секцию типа  $S(X)$  здесь не инициируют поиск соответствующих данных в сети, а лишь создают блокировку, ожидая завершения ранее инициированной (операцией типа  $F(X)$ ) передачи данных в случае если передача к данному моменту еще не успела завершиться.

За внешним равноправием и взаимозаменяемостью узлов системы был расположен слой абстракции, вводящий определённую специализацию, а именно – понятие роли, которую узел выполняет в системе. Соответственно был разработан алгоритм ролей и переходов, назначение которого – соответствовать условиям высокой динамичности системы, обеспечивая её устойчивость к сбоям отдельных узлов. Схематично роли и возможность их изменения представлены

на рисунке 2.

Расшифровка названий ролей и краткое пояснение к ним приведены в таблице 1.

Так как МАКС DSM является динамической системой, роли узлов не могут быть фиксированными. В связи с этим каждый узел потенциально может выполнять любую роль, а также менять её в процессе работы, за единственным исключением – оставив однажды роль Новичок, вернуться к этой роли узел уже не сможет<sup>1</sup>.

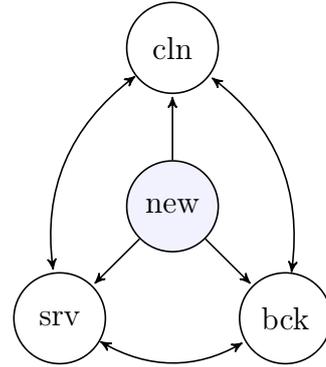


Рисунок 2 – Схема возможных изменений ролей узла в МАКС DSM

Название	Сокращение	Описание
Новичок	new	Новый узел, роль не определена
Сервер	srv	Главный узел в системе
Копия	bck	Резервная копия сервера
Клиент	cln	Обычный узел

Таблица 1 – Роли узлов в МАКС DSM

Условия, в которых узел может изменить свою роль, выражены алгоритмом на рисунке 3. Для удобства, названия процедур отражают текущую роль узла. Значения переменных  $x_1$ ,  $x_2$ ,  $x_3$  определяются низлежащим сетевым уровнем и зависят от свойств конкретных каналов.

В дополнение к алгоритму выше, в данной главе были рассмотрены вопросы обнаружения сбоев в системе, такие как потеря связи в процессе передачи группы низкоуровневых сообщений, а также построена модель обмена сообщениями в системе и отмечена необходимость проработки реакции системы на отказ произвольного узла в каждом из возможных состояний системы.

Анализ особенностей ранее созданных DSM систем, отрицательно сказывающихся на их популярности, позволил выработать концепцию (принципы и

<sup>1</sup>Ситуации подобно перезагрузке узла мы рассматриваем как исчезновение узла и возникновение нового.

---

```

1: procedure НОВИЧОК
2:   if продолжительность тишины в эфире  $> x_1$  then
3:     роль  $\leftarrow$  Сервер
4:   else if у Сервера нет Копии then
5:     роль  $\leftarrow$  Копия
6:   else
7:     роль  $\leftarrow$  Клиент
8: procedure КЛИЕНТ
9:   if продолжительность «молчания» Сервера  $> x_2$  then
10:    роль  $\leftarrow$  Сервер
11:  else if сервер требует стать Копией then
12:    роль  $\leftarrow$  Копия
13: procedure КОПИЯ
14:   if продолжительность «молчания» Сервера  $> x_3$  then  $\triangleright x_3 > x_2$ 
15:    роль  $\leftarrow$  Сервер
16: procedure СЕРВЕР
17:   if нет Копии И есть Клиенты then
18:     потребовать от произвольного Клиента стать Копией

```

---

Рисунок 3 – Алгоритм смены роли узлом в МАКС DSM

требования) к прикладному интерфейсу, свободному от выявленных недостатков. Основными требованиями стали реализация на стандартном языке Си++, отсутствие требований прикладному программисту по изучению новых техник программирования или средств разработки, а также контроль соответствия требованиям модели на этапе компиляции.

Таким образом, в данной главе были решены все поставленные во введении теоретические задачи: предложены усиленная модель консистентности по выходу, алгоритм ролей и переходов, а также сформулирована концепция прикладного интерфейса (в виде требований). На основе полученных результатов в следующей главе рассматривается программная реализация создаваемого решения МАКС DSM, включая разработку интерфейса прикладного взаимодействия с данной системой, результаты проведённых измерений производительности.

**В третьей главе** описана реализация прикладного интерфейса, принципы системы сообщений и ключевые фрагменты протокола, рассмотрена реали-

зация отказоустойчивости, а также представлен пример программы, функционирующей на нескольких устройствах и демонстрирующей все возможности созданной системы. В завершение рассматривается испытательный стенд и приводятся результаты измерений производительности решения.

Использование техники метапрограммирования (посредством языка C++ и директив препроцессора, что отличается от ранее применявшихся при реализации DSM систем подходах), позволило достичь сочетания в одном решении качеств, ранее совместно не встречавшихся: простоты в использовании, проверки корректности на этапе компиляции и отсутствия зависимости от специфичных компонентов среды разработки (таких как специализированный для конкретной DSM системы компилятор или препроцессор). Ожидается, что данные свойства будут способствовать широкому распространению разработанного решения.

Система ролей позволила выстроить лаконичный протокол обмена сообщениями (каждое из которых может быть лишь одного из 16-ти типов), что повышает надежность реализации решения. Дополнительно был введен принцип атомарности высокоуровневых сообщений (транзакций): высокоуровневая транзакция уровня DSM, состоящая из множества низкоуровневых сообщений, не вызывает изменения состояния системы, если в процессе передачи одного из низкоуровневых сообщений произошел сбой. Данный подход позволил снизить сложность реализации требования отказоустойчивости, так как предоставил возможность оперировать только высокоуровневыми понятиями. Отказоустойчивость на уровне DSM реализуется рассмотрением всех возможных ситуаций выхода из строя устройств любой роли, в том числе одновременного выхода из строя нескольких устройств. Для каждой ситуации предусмотрена процедура восстановления системы. Хотя восстановление системы после сбоя отдельного узла или нескольких узлов происходит достаточно быстро, следует отметить, что в текущей реализации не обрабатывается ситуация, когда сбой происходит в процессе восстановления системы от предыдущего сбоя (каскадный сбой) – данный недостаток необходимо вынести на дальнейшую проработку вне рамок текущего исследования.

На верхнем уровне узел МАС, функционирующий под управлением ОСРВ МАКС с интегрированным решением МАКС DSM можно представить в виде,

изображенном на рис. 4. Диаграмма основных классов МАКС DSM в нотации UML представлена в разделе 3.5.2. диссертации.

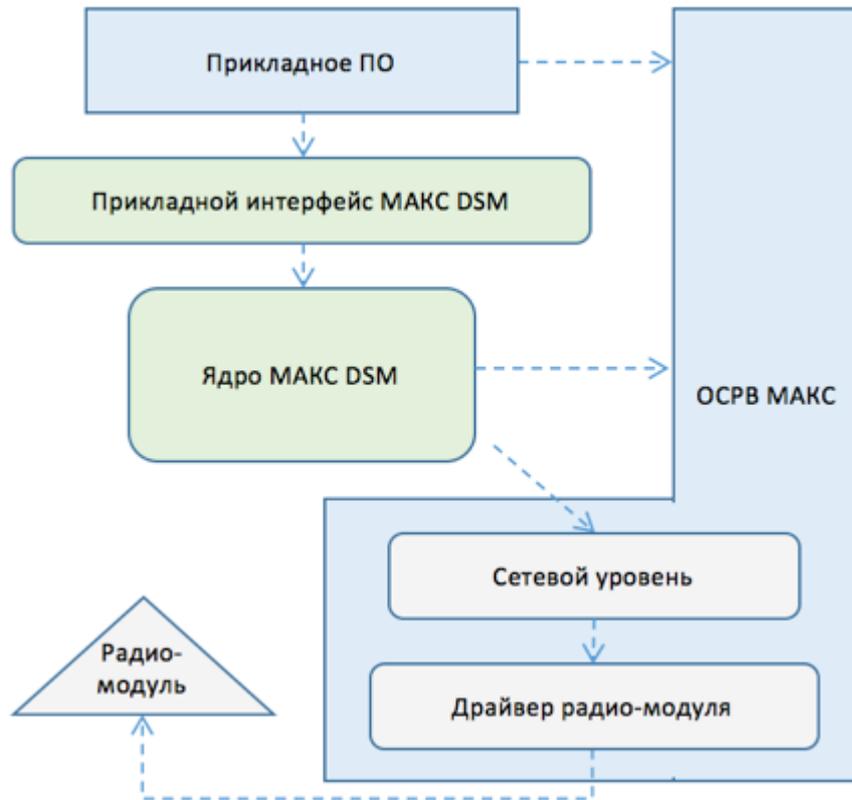


Рисунок 4 – МАКС DSM в составе функционирующего узла МАКС

В данной главе также рассмотрена экспериментальная распределённая программа, выполняющая вычислительную задачу на нескольких устройствах. Показано, что благодаря МАКС DSM, исходный код программы оказывается крайне лаконичен, обеспечивая при этом синхронную работу нескольких устройств и работоспособность системы в случае выключения отдельных узлов. Таким образом, работоспособность созданного решения была доказана.

Глава завершается описанием проведенных измерений – как на реальном, специально созданном оборудовании (для системы от двух до пяти устройств, рис. 5), так и на модели (от двух до 16 устройств).

Для более глубокого понимания происходящих в системе процессов и более качественной интерпретации результатов, аналогичные представленным выше измерения были проведены для всех логических уровней сетевых протоколов, полученные результаты – исследованы и объяснены. Введение уровня МАКС DSM снижает производительность системы (относительно низлежащего сетево-

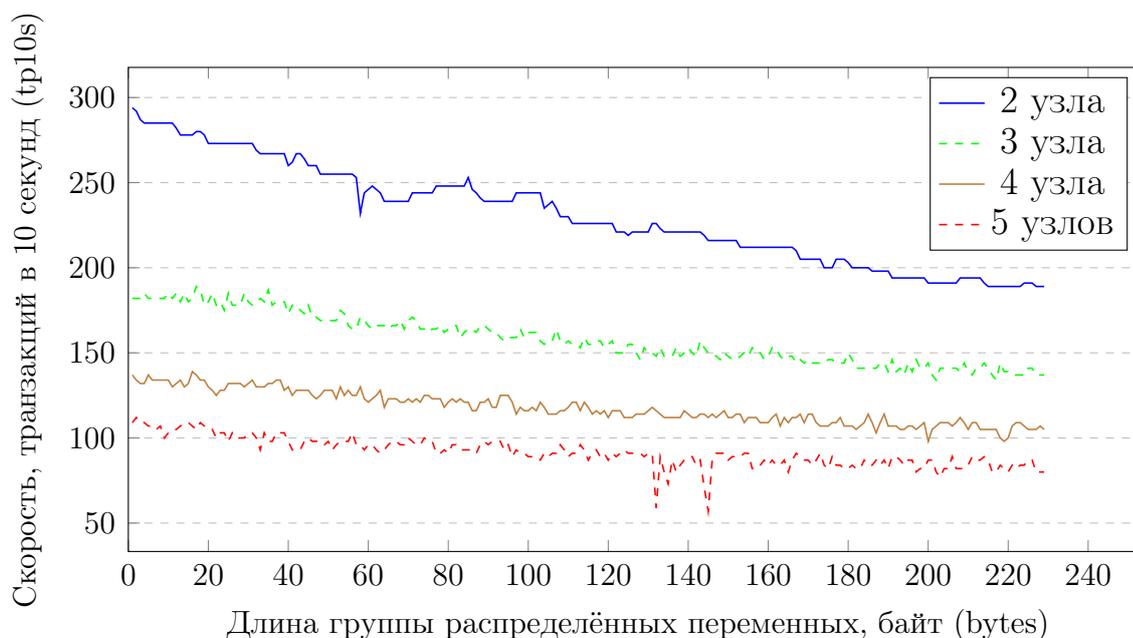


Рисунок 5 – Скорость DSM уровня (2–5 узлов)

го уровня) приблизительно в десять раз, что представляет собой ожидаемый результат и, в основном, является следствием количества сообщений, необходимых для выполнения отдельной операции (транзакции) данного уровня (включая подтверждения о доставке сообщений).

## Заключение

Целью данного диссертационного исследования была разработка моделей, алгоритмов и программных средств, реализующих концепцию распределённой общей памяти для мультиагентных систем в IoT и позволяющих существенно упростить и ускорить создание прикладных решений в данной области. Заявленная цель была достигнута.

**Итоги выполненного исследования.** Основные результаты диссертационной работы заключаются в следующем.

1. Разработана усиленная модель консистентности по выходу, позволяющая гарантировать согласованность данных в различных узлах распределённой мультиагентной системы и отличающаяся от существующих тем, что сочетает в себе свойства нескольких других моделей, в своей совокупности позволяющие минимизировать время существования уникальных данных в системе и обеспечить высокий уровень устойчивости к непреднамеренным ошибкам прикладного программиста, вместе с тем, не наследуя свойств, избыточных для

обозначенной сферы применения.

2. Разработан алгоритм организации узлов мультиагентной системы в самовосстанавливающуюся структуру, устойчивую к выходу из строя отдельных узлов. Алгоритм основан на концепции ролей – каждый узел системы исполняет некоторую роль, динамически сменяющуюся в случае изменений в конфигурации системы. Проработаны возможные аварийные ситуации, способы их выявления и восстановления системы без потери функциональности и данных. Также определена ситуация, в которой восстановление системы в текущей реализации не обеспечивается (каскадный отказ нескольких узлов).

3. Разработаны принципы программного интерфейса для прикладного взаимодействия с созданным механизмом реализации концепции распределённой памяти. При формировании принципов были учтены недостатки прошлых решений, препятствующие широкому распространению соответствующих систем.

4. Разработано программное обеспечение, реализующее концепцию распределённой общей памяти для мультиагентных систем в сфере IoT. Созданное решение позволяет существенно упростить задачу организации взаимодействия устройств в МАС, что показано на примере использования данного решения.

5. Создан экспериментальный программно-аппаратный стенд из пяти устройств. С его помощью собраны характеристики решения для конфигураций от двух до пяти устройств в системе. В дополнение к стенду создан программный имитационный комплекс, позволяющий предсказать характеристики решения в более широких пределах вариантов использования. Достигнута точность модели, обеспечивающая сравнимые характеристики модели и программно-аппаратного стенда при одинаковом количестве устройств. На модели произведены замеры для количества устройств от двух до шестнадцати.

**Рекомендации по применению результатов работы.** При применении результатов данной работы в научных исследованиях или на производстве необходимо учитывать следующие аспекты.

1. При переносе МАКС DSM на другие платформы и среды разработки необходимо учитывать возможности соответствующего компилятора. Система МАКС DSM реализована на языке C++ в рамках стандарта ISO/IEC 14882:2003, однако в области встраиваемых решений поддержка даже распро-

странённых стандартов может быть ограничена.

2. При использовании решения необходимо учитывать, что устойчивость к каскадным сбоям, когда очередной сбой происходит в процессе восстановления системы от предыдущего сбоя, в данный момент решением не обеспечивается.

3. При использовании системы в практических задачах с целью максимизации производительности следует настраивать величину таймаутов в алгоритме смены роли узлом в соответствии с характеристиками конкретной сети.

**Перспективы дальнейшей разработки темы.** В ходе проведения исследования было выявлено несколько моментов, нуждающихся в дополнительной проработке вне рамок текущей работы. Основные из них перечислены ниже.

1. Доработка решения в плане отказоустойчивости – необходимо предусмотреть устойчивость к каскадным сбоям, в ситуациях когда очередной сбой происходит в процессе восстановления системы от предыдущего сбоя.

2. Сбор метрик решения, связанных со скоростью восстановления системы после сбоев отдельных узлов.

3. Выявление причин аномалий, обнаруженных при проведении замеров на оборудовании. Данные аномалии выглядят незначительными, однако потенциально могут быть следствием неучтённых, но существенных явлений.

4. Оптимизация решения по производительности, сравнение вариантов по всем метрикам, включая скорость восстановления системы после сбоев. К примеру, одним из способов существенной оптимизации может стать ликвидация роли «Копия» – необходимо оценить влияние данного решения на скорость восстановления и возможности масштабирования решения.

5. Разработка моделей программирования, примеров для распространённых вариантов использования. Например, конечное ПО, предназначенное для группы устройств, порождающих новые данные, может существенно отличаться от ПО для системы, в которой дополнительные устройства лишь резервируют основной узел. В обоих случаях МАКС DSM позволяет существенно упростить разработку конечного решения, однако для достижения оптимального результата нужно придерживаться различных принципов разработки.

## Список публикаций автора по теме диссертации

В журналах из перечня рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, доктора наук

1. Бойко, П. В. МАКС DSM: Метаязык для организации взаимодействия групп автономных аппаратов [Текст] / П. В. Бойко // Навигация и гидрография. — 2017. — № 50. — С. 7–11.
2. Бойко, П. В. Подход к задаче обеспечения когерентности распределённых данных в мультиагентной системе [Текст] / П. В. Бойко // Инновации и инвестиции. — 2017. — № 2. — С. 206–208.
3. Бойко, П. В. Разработка прикладного API системы распределенной общей памяти МАКС DSM [Текст] / П. В. Бойко // Системный администратор. — 2017. — № 6. — С. 12–13.
4. Бойко, П. В. Распределённая общая память как способ организации взаимодействия в мультиагентных системах [Текст] / П. В. Бойко // Инновации и инвестиции. — 2017. — № 3. — С. 113–117.
5. Бойко, П. В. DSM в IoT: Усиленная модель консистентности по выходу [Текст] / П. В. Бойко // Инновации и инвестиции. — 2017. — № 7. — С. 134–136.
6. Бойко, П. В. ОСПВ МАКС (операционная система реального времени для мультиагентных когерентных систем) [Текст] / П. В. Бойко. — [Б. м.] : Свидетельство о государственной регистрации программы для ЭВМ № 2016617143, 28.06.2016 (Роспатент).