

**Федеральное государственное
бюджетное образовательное учреждение высшего образования
«Воронежский государственный технический университет»**

На правах рукописи



РЫКШИН Максим Сергеевич

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ПРОЦЕССОВ УПРАВЛЕНИЯ МНОГОАГЕНТНЫМ
МОНИТОРИНГОМ ОБЪЕКТОВ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ
ПРИ НЕСТАЦИОНАРНОЙ НАГРУЗКЕ**

Специальность 2.3.5. Математическое и программное обеспечение
вычислительных систем, комплексов и
компьютерных сетей

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук
Кравец Олег Яковлевич

Воронеж – 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1. Проблемы функционирования распределенных систем мониторинга при возникновении нестационарной нагрузки на объектах мониторинга.....	12
1.1 Многоуровневая архитектура распределенных вычислительных систем и проблематика их масштабирования.....	12
1.2 Архитектура систем мониторинга распределенных вычислительных систем и проблематика нестационарности нагрузки на них.....	21
1.3 Анализ методов моделирования процесса мониторинга при нестационарной нагрузке.....	34
1.4 Формальная постановка задачи исследования.....	43
1.5 Выводы по главе.....	45
ГЛАВА 2. Разработка математического обеспечения распределенной системы мониторинга для прогнозирования моментов нестационарной нагрузки, имеющей временной лаг распространения.....	48
2.1 Исследование подходов к организации децентрализованной архитектуры распределенной системы мониторинга на основе интеллектуальных агентов мониторинга.....	48
2.1.1 Анализ структурной и функциональной моделей систем мониторинга с централизованной структурой.....	48
2.1.2 Анализ структурной и функциональной моделей систем мониторинга с децентрализованной структурой на основе мультиагентных систем.....	52
2.2 Разработка нейросетевой модели интеллектуального агента мониторинга, на основе рекуррентной нейронной сети с длинной цепью элементов краткосрочной памяти.....	57
2.2.1 Выбор варианта нейросетевой модели интеллектуального агента мониторинга.....	58

2.2.2 Структура нейросетевой модели интеллектуального агента мониторинга для решения задачи прогнозирования моментов возникновения нестационарной нагрузки	66
2.3 Моделирование процесса обучения разработанной нейросетевой модели интеллектуального агента мониторинга	70
2.4 Выводы по главе.....	73
ГЛАВА 3. РАЗРАБОТКА ОБОБЩЕННОГО АЛГОРИТМА ДИНАМИЧЕСКОЙ НАСТРОЙКИ ПАРАМЕТРОВ РАЗРАБОТАННОЙ НЕЙРОСЕТЕВОЙ МОДЕЛИ В ЗАДАННОЙ ТОЧКЕ МОНИТОРИНГА И ОБОБЩЕННОГО АЛГОРИТМА ВЗАИМНОГО ИНФОРМАЦИОННОГО СОГЛАСОВАНИЯ ИАМ.....	75
3.1 Разработка обобщенного алгоритма динамической настройки параметров разработанной нейросетевой модели в заданной точке мониторинга	75
3.2 Разработка обобщенного алгоритма взаимного информационного согласования интеллектуальных агентов мониторинга.....	80
3.2.1 Разработка алгоритма управления многоадресной рассылкой сообщений.....	81
3.2.2 Разработка алгоритма наложенного управления сообщениями	84
3.3 Выводы по главе.....	97
ГЛАВА 4. РАЗРАБОТКА АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОЙ МНОГОАГЕНТНОЙ СИСТЕМЫ МОНИТОРИНГА ОБЪЕКТОВ МУЛЬТИСЕРВИСНОЙ СЕТИ СВЯЗИ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ МОНИТОРИНГА.....	98
4.1 Разработка архитектуры распределенной многоагентной системы мониторинга объектов мультисервисной сети связи	98
4.2 Разработка программного обеспечения интеллектуальных агентов мониторинга.....	101

4.3 Проведение численного эксперимента для оценки эффективности полученных решений.....	105
4.4 Выводы по главе.....	114
ЗАКЛЮЧЕНИЕ	115
СПИСОК ТЕРМИНОВ, СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ..	117
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	119

ВВЕДЕНИЕ

Актуальность темы. Развитие современных распределенных систем различной функциональности связано, как с увеличением объема информации, используемой для их эффективного функционирования, так и с увеличением сложности вычислительных процессов, необходимых для обработки этой информации, что требует решения целого ряда задач в различных предметных областях.

Одной из наиболее важных задач, является информационный мониторинг состояния узлов распределенных систем. Для решения этой задачи используются системы мониторинга (СМ), например, интегрированная система управления сетью связи (Integrated network management system, INMS), система управления центром обработки данных (Data Center Infrastructure Management, DCIM), автоматизированные системы управления дорожным движением (АСУДД) и ряд других. Архитектура современных реализаций СМ оверлейно развертывается поверх узлов объекта мониторинга и основывается на централизованной модели «агент-менеджер» с единым или многоуровневым центром управления. При этом узлы-агенты таких СМ выполняют только функции сбора параметров объекта мониторинга с заданной скважностью рабочего цикла и их передачи на узлы-менеджеры, реализующие формирование управляющего воздействия. Подобное архитектурное решение имеет существенные ограничения в условиях динамически возникающей нестационарной нагрузки на отдельные узлы или группы узлов объекта мониторинга. Примером таких объектов мониторинга являются мультисервисные сети связи, поддерживающие технологии «интернет вещей», «умный город», системы беспилотного транспорта, которые на коротких временных периодах могут формировать пиковые значения сетевого трафика. При этом для крупномасштабных, территориально распределенных объектов мониторинга характерно наличие временного лага (time lag) распространения нестационарной нагрузки по их узлам, который также не учитывается существующим контуром централизованного управления СМ.

Одним из вариантов решения указанных проблем СМ является использование децентрализованной одноранговой многоагентной архитектуры, основанной на интеллектуальных агентах – узлах СМ, решающих не только задачу сбора параметров о закреплённом объекте мониторинга, но и их анализа, а также формирования управляющего воздействия. Прогнозирование моментов возникновения пиковой (аномальной) нагрузки, с учетом временного лага ее распространения по узлам объекта мониторинга, при этом может осуществляться за счет их согласованного информационного взаимодействия.

Существенный вклад в развитие предметной области МАС, в частности математического и программного обеспечения для систем информационного мониторинга, внесли А.А. Петров, В.И. Петренко, В.В. Афанасьев, А.И. Тихомиров, М.И. Вексельман, Р.Р. Галимов, М. Zachary, J. Blakley, A. Ezzat, M. Alizadeh, D. Reed, L Jong, J. Vicente, N. Masinde.

Следует отметить, что большинство исследований посвящено либо решению задачи балансировки вычислительной нагрузки между интеллектуальными агентами, либо ориентировано на узко специализированные предметные области, например, автономные робототехнические системы.

Таким образом, актуальность темы диссертационного исследования связана с тем, что современные распределенные системы требуют решения ряда задач по разработке дополнительных средств математического и программного обеспечения многоагентных вычислительных систем, обеспечивающих функции мониторинга в нестационарных условиях.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

Целью работы является разработка математического и программного обеспечения для повышения эффективности процесса функционирования распределённой системы мониторинга при нестационарной нагрузке на объекте мониторинга.

Задачи исследования. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ состояния проблемы мониторинга объектов различного класса, функционирующих в условиях нестационарной нагрузки, которая имеет временной лаг распространения, с целью выявления общих принципов процесса их мониторинга.

2. Разработать нейросетевую модель интеллектуального агента системы мониторинга для узла объекта мониторинга, обеспечивающую решение задачи прогнозирования нагрузки на узле мониторинга на основе анализа временного ряда ее распространения.

3. Разработать обобщенный алгоритм настройки входных параметров нейронной сети интеллектуального агента, динамически подстраивающий их в зависимости от параметров нестационарной нагрузки, имеющей временной лаг распространения.

4. Разработать обобщенный алгоритм взаимного информационного согласования множества интеллектуальных агентов в распределенной многоагентной системе мониторинга.

5. Разработать архитектуру программно реализованной распределенной многоагентной системы мониторинга.

6. Провести численные эксперименты по оцениванию качества функционирования разработанной распределенной многоагентной системы мониторинга на примере мониторинга узлов фрагмента транспортной сети с коммутацией пакетов для различных условий возникновения нестационарной нагрузки, имеющей временной лаг распространения.

Объект исследования: процессы управления в условиях распределённой системы мониторинга.

Предмет исследования: математические методы управления мониторингом при нестационарной нагрузке, имеющей временной лаг распространения.

Методы исследования. При решении поставленных в диссертации задач использовались методы теории системного анализа, методы машинного обучения,

теория нейронных сетей, теория многоагентных систем, методы имитационного моделирования, математической статистики и планирования экспериментов.

Тематика работы соответствует следующим пунктам паспорта специальности 2.3.5. Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей (технические науки): п.3 «Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем»; п.9 «Модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных».

Научная новизна работы. В диссертации получены следующие результаты, характеризующиеся научной новизной:

- нейросетевая модель интеллектуального агента распределенной системы мониторинга, отличающаяся от известных применением структуры рекуррентной нейронной сети с длинной цепью элементов краткосрочной памяти и обеспечивающая возможность формирования матрицы связности гиперпараметров нейронной сети и накопленных значений параметров состояния объектов мониторинга с учетом характеристик временного ряда распространения нестационарной нагрузки;

- обобщенный алгоритм динамической настройки параметров нейронной сети в заданной точке мониторинга, отличающийся от известных использованием марковского процесса принятия решения в задаче обучения с подкреплением на основе модификации алгоритма итерации по значениям, обеспечивающий минимизацию временных затрат на обучение при заданном значении точности прогнозирования;

- обобщенный алгоритм взаимного информационного согласования интеллектуальных агентов, отличающийся от известных учетом пространственно-временной корреляции данных, порождаемых множеством интеллектуальных агентов, обеспечивающий сокращение времени задержки передачи данных о моментах возникновения нестационарной нагрузки соседним интеллектуальным

агентам, расположение которых соответствует направлению распространения временного лага нагрузки;

- архитектура распределенной многоагентной системы мониторинга объектов мультисервисной сети связи, отличающаяся от известных реализацией двухуровневой схемы вычислительной инфраструктуры и обеспечивающая поддержание логической целостности данных в процессе взаимного информационного согласования подмножества входящих в нее интеллектуальных агентов.

Теоретическая значимость исследования состоит в разработке нового обобщенного алгоритма взаимного информационного согласования агентов многоагентных систем, использующих структуру рекуррентной нейронной сети с длинной цепью элементов краткосрочной памяти, позволяющего достичь равновесного состояния информационной осведомленности подмножества агентов о прогнозном значении пиковой нагрузки на объектах мониторинга. Положения и выводы, содержащиеся в данной работе, могут быть использованы в развитии программных средств перспективных систем мониторинга.

Практическая значимость. Реализация разработанного обобщенного алгоритма взаимного информационного согласования интеллектуальных агентов в виде специализированного программного средства с модульной структурой, позволяет интегрировать его в существующие программные реализации систем информационного мониторинга. На специализированное программное средство получено свидетельство о государственной регистрации.

Положения, выносимые на защиту:

- нейросетевая модель интеллектуального агента системы мониторинга обеспечивает формирование матрицы связности гиперпараметров нейронной сети и накопленных значений параметров состояния объектов мониторинга с учетом характеристик временного ряда распространения нестационарной нагрузки;

- обобщенный алгоритм динамической настройки параметров нейронной сети в заданной точке мониторинга минимизирует временные затраты на обучение при заданном значении точности прогнозирования;

- обобщенный алгоритм взаимного информационного согласования интеллектуальных агентов в распределенной многоагентной системе мониторинга обеспечивает сокращение времени задержки передачи данных о моментах возникновения нестационарной нагрузки соседним интеллектуальным агентам, расположение которых соответствует направлению распространения временного лага нагрузки;

- архитектура распределенной многоагентной системы мониторинга объектов мультисервисной сети связи, за счет двухуровневой схемы, поддерживает логическую целостность данных в процессе взаимного информационного согласования подмножества входящих в нее интеллектуальных агентов.

Результаты внедрения. Основные положения и результаты диссертационной работы реализованы в виде специального программного средства управления многоагентным мониторингом объектов распределенной системы при нестационарной нагрузке и нашли практическое применение в технологической компании ООО «Айти Интегра Системс», а также внедрены в образовательный процесс Академии ФСО России (дисциплина – «Информатика»).

Апробация результатов диссертационного исследования. Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: XXV-th International Open Science Conference «Modern informatization problems in simulation and social technologies (MIP-2020'AS)» (USA, 2020), XXVI-th International Open Science Conference «Modern informatization problems in the technological and telecommunications systems analysis and synthesis (MIP-2021'AS)» (USA, 2021), XII Всероссийская межведомственная научная конференция «Актуальные направления развития систем охраны, специальной связи и информации для нужд ОГВ РФ» (Академия ФСО России, Орел, 2021 г.), а также на научных семинарах кафедры автоматизированных и вычислительных систем ВГТУ (2020-2023 гг.).

Публикации. По результатам диссертационного исследования опубликовано 9 печатных работ, в т.ч. 4 статьи в изданиях, рекомендованных ВАК РФ, свидетельство о государственной регистрации программы для ЭВМ, а также

статья в издании, индексируемом в Scopus. В работах, опубликованных в соавторстве и приведённых в конце автореферата, лично автором получены следующие результаты: [64] - нейросетевая модель интеллектуального агента системы мониторинга, [71,72] - обобщенный алгоритм динамической настройки параметров нейронной сети в заданной точке мониторинга, [73] – обобщенный алгоритм взаимного информационного согласования интеллектуальных агентов в распределенной многоагентной системе мониторинга, [77,78] - архитектура распределенной многоагентной системы мониторинга объектов мультисервисной сети связи.

Объем и структура работы. Диссертация состоит из введения, четырех глав и заключения. Работа изложена на 127 страницах машинописного текста, включая 67 рисунков, 4 таблиц и список литературы из 86 наименований.

ГЛАВА 1. ПРОБЛЕМЫ ФУНКЦИОНИРОВАНИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ МОНИТОРИНГА ПРИ ВОЗНИКНОВЕНИИ НЕСТАЦИОНАРНОЙ НАГРУЗКИ НА ОБЪЕКТАХ МОНИТОРИНГА

В данной главе представлена проблематика процесса функционирования систем мониторинга высоконагруженных распределенных вычислительных систем различного класса в условиях возникновения на их отдельных узлах нестационарной нагрузки, которая имеет временной лаг распространения.

Рассмотрены архитектурные особенности подобных систем, архитектурные особенности систем мониторинга в различных предметных областях, и обобщены проблемы их функционирования применительно к условиям возникновения на объекте мониторинга нестационарной нагрузки, распространяющейся по группам узлов мониторинга с некоторым временным лагом.

В главе в обобщенном описаны методы информационного мониторинга, в частности в моменты возникновения нестационарной нагрузки на узлах мониторинга, а также технологические аспекты их реализации. Формулируется цель и делается постановка научной задачи исследования

1.1 Многоуровневая архитектура распределенных вычислительных систем и проблематика их масштабирования

Развитие и совершенствование современных распределенных вычислительных систем различного класса связано, в первую очередь, с потенциальным усложнением их архитектурных особенностей, обусловленным, с одной стороны, необходимостью масштабирования их функциональных узлов для обеспечения обслуживания высокого уровня нагрузки, порождаемой потребителями функций этих систем, а с другой – усложнением самих функций этих узлов.

Показательным примером проблематики комплексного масштабирования высоконагруженных распределенных вычислительных систем является развитие

инфраструктуры IoT-систем – специализированных распределенных вычислительных систем, терминальной базой которых являются подключенные интеллектуальные устройства «Интернета вещей» (IoT – Internet of Things) различного уровня функциональности [1]. В общем виде многоуровневая организация подобных систем представлена на рисунке 1.1.

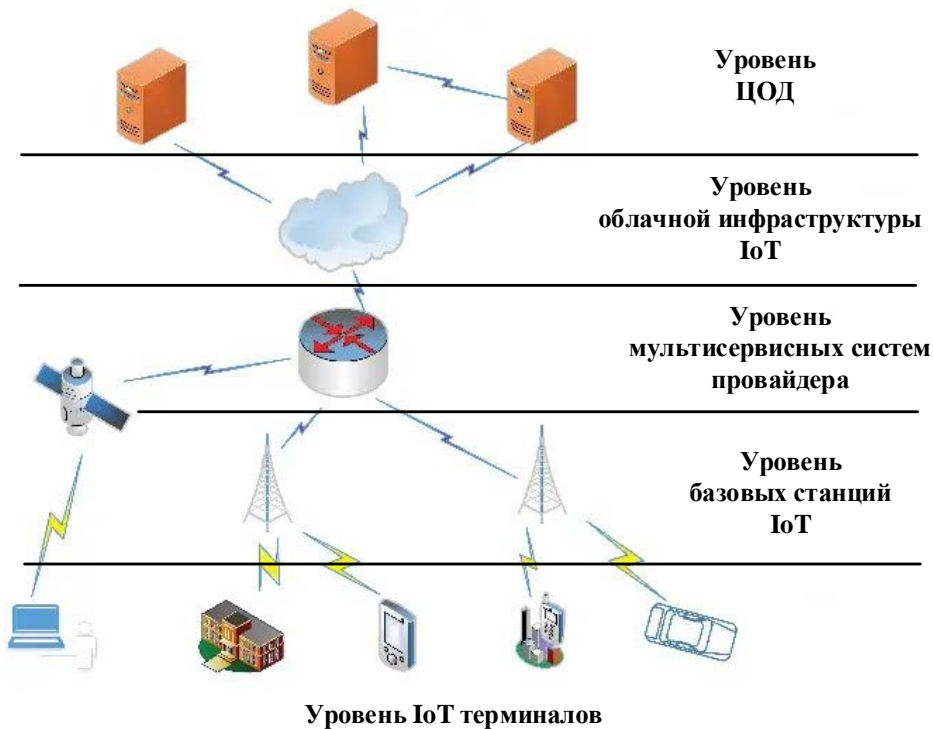


Рисунок 1.1 – Обобщенная многоуровневая организация распределенной вычислительной IoT-системы

Сложность организации крупномасштабных распределенных вычислительных систем, к которым можно отнести различные реализации IoT-систем, обусловлена, в первую очередь их структурной и функциональной сложностью, требующей разделения на соответствующие структурные и функциональные уровни (рисунок 1.1). На каждом из этих уровней, в свою очередь функционирует частный вариант специализированной распределенной системы, решающей задачи, соответствующие ее функциональному предназначению.

Для конкретизации рассмотрим вариант реализации распределенной вычислительной IoT-системы, связанной с интеллектуальным управлением транспортом.

1. На уровнях IoT-терминалов и базовых станций функционируют интеллектуальные транспортные системы (Intelligent Transportation Systems (ITS)) – системы управления, предназначенные для автоматизированного управления транспортно-дорожным комплексом (ТДК) некоторого региона, конкретным транспортным средством или группой транспортных средств с целью обеспечения максимизации показателей использования дорожной сети [2,3]. В обобщенном виде структура подобных систем определена в [4] (рисунок 1.2). Наглядное представление базовых элементов ITS дается на рисунке 1.3.

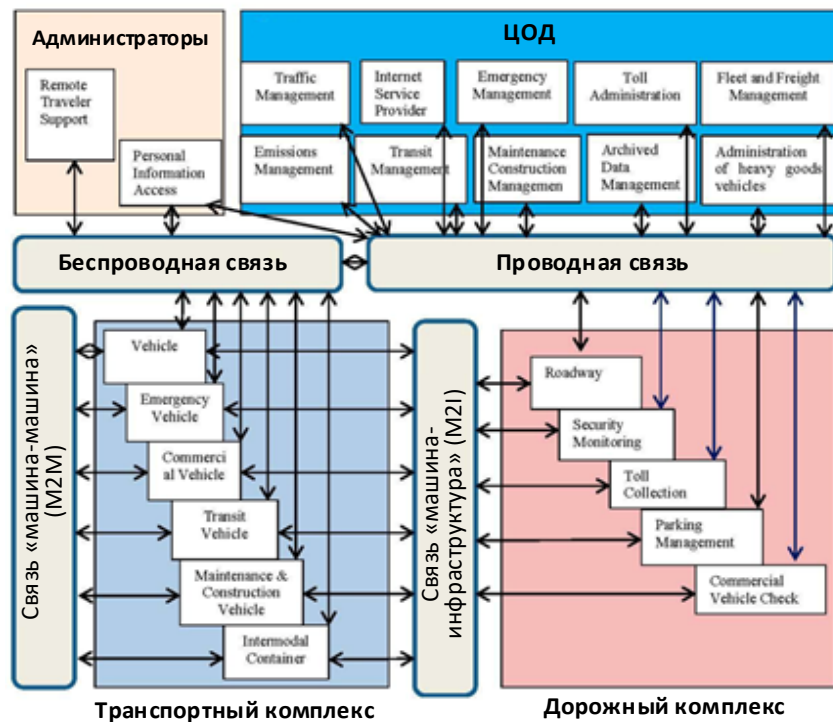


Рисунок 1.2 – Обобщенная схема распределенной вычислительной системы - интеллектуальная транспортная система

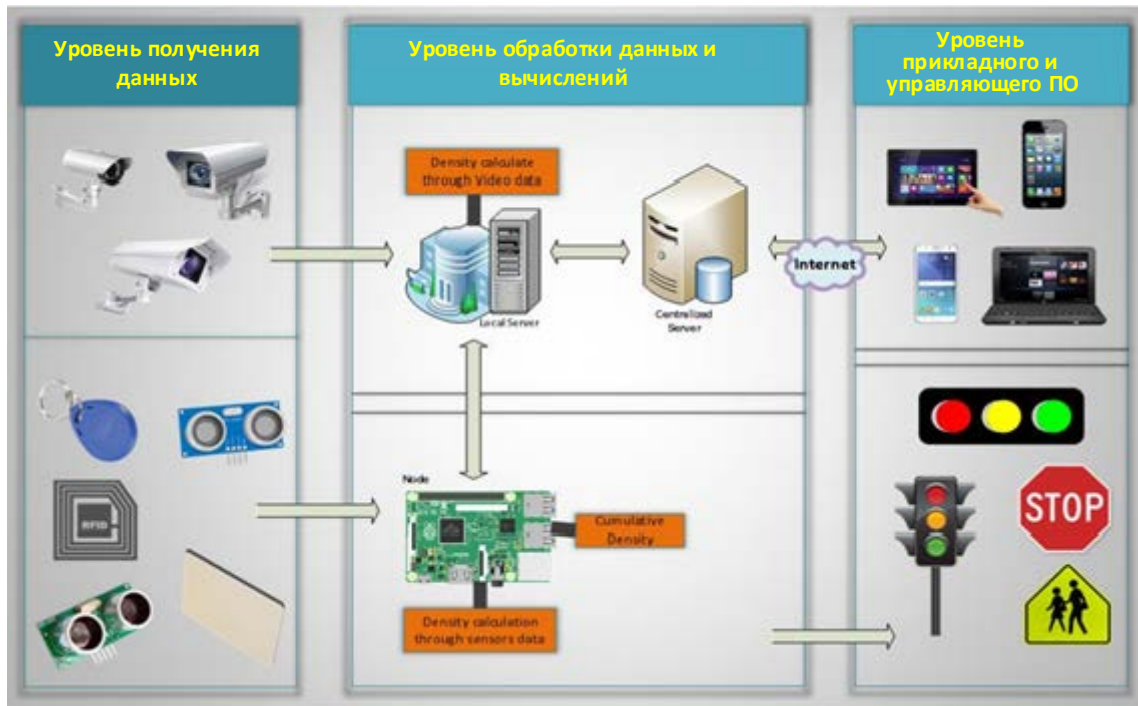


Рисунок 1.3 – Многоуровневое представление базовых компонентов интеллектуальной транспортной системы

Из рисунков 1.2 и 1.3 видно, что в общем случае системы ITS охватывают процессы управления тремя компонентами: дорожной инфраструктурой, транспортной инфраструктурой и пользователями ТДК. Масштаб подобных систем существенно зависит от каждого из этих компонентов и крупных городах является переменной величиной, связанной с постоянным усложнением ТДК.

2. На уровне мультисервисных систем провайдеров функционируют мультисервисные системы связи (МСС) - многоцелевые распределенные вычислительные инфраструктуры, предназначенные для передачи речи, изображений и данных и базирующиеся на принципах пакетной коммутации [5]. Основная задача МСС заключается в том, чтобы взаимодействие компонентов выше и ниже лежащих уровней в единой транспортной среде. Архитектура МСС обобщает как сетевую составляющую, так и серверные и терминальные вычислительные системы, предоставляющие услуги потребителям. Обобщенная архитектура МСС показана на рисунке 1.4.

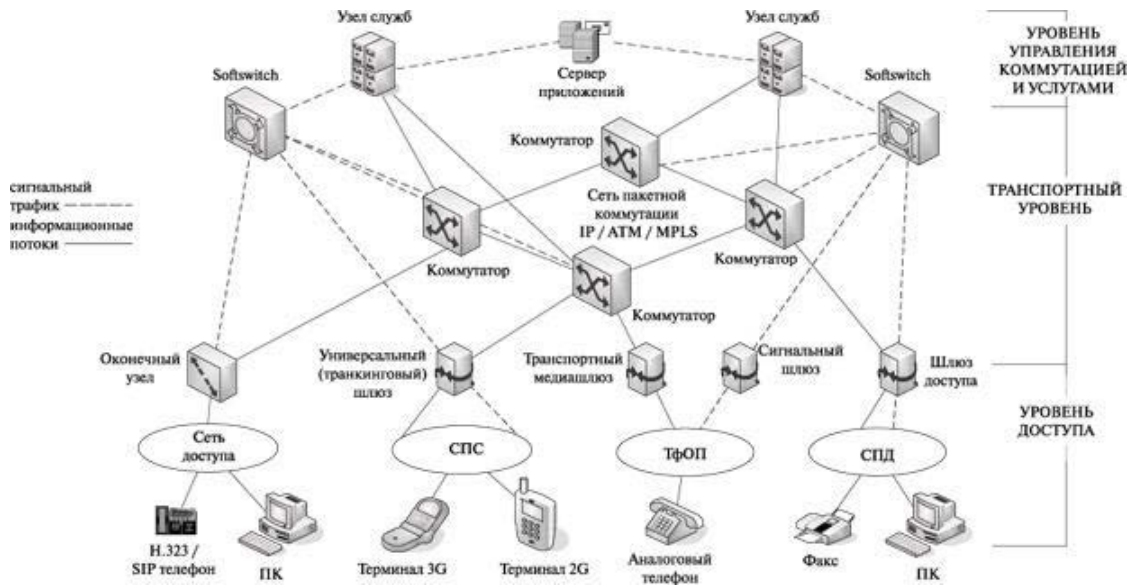


Рисунок 1.4 – Обобщенная схема мультисервисной системы связи

Проблема, связанная с масштабированием МСС, основана, как на росте потребительской базы, так на развитии возможностей самих сервисов.

3. На уровнях облачной инфраструктуры и центров обработки данных (ЦОД), представлены распределенные вычислительные инфраструктуры, основанные на технологиях виртуализации вычислительных ресурсов, систем хранения данных (СХД) и программно-определяемых сетей. Для эффективной реализации предоставляемых потребителям информационных услуг в рамках подобных ЦОД динамически формируются такие ИТ-инфраструктуры, как IaaS, SaaS, PaaS [6, 7]. Их конфигурация в разные моменты времени существенно зависит, как от интенсивности поступающих пользовательских запросов, так и от объема и интенсивности поступления обрабатываемых массивов данных. В настоящее время, в связи с ростом популярности информационных систем, базирующихся на генеративных нейронных сетях [8,9], нагрузка подобных ЦОД существенно возросла. При этом, как с точки зрения интенсивности пользовательских запросов, так и с точки зрения получения массивов данных из внешних источников, необходимых для формирования обучающих выборок (датасетов).

Обобщенная схема подобных ЦОД представлена на рисунке 1.5.

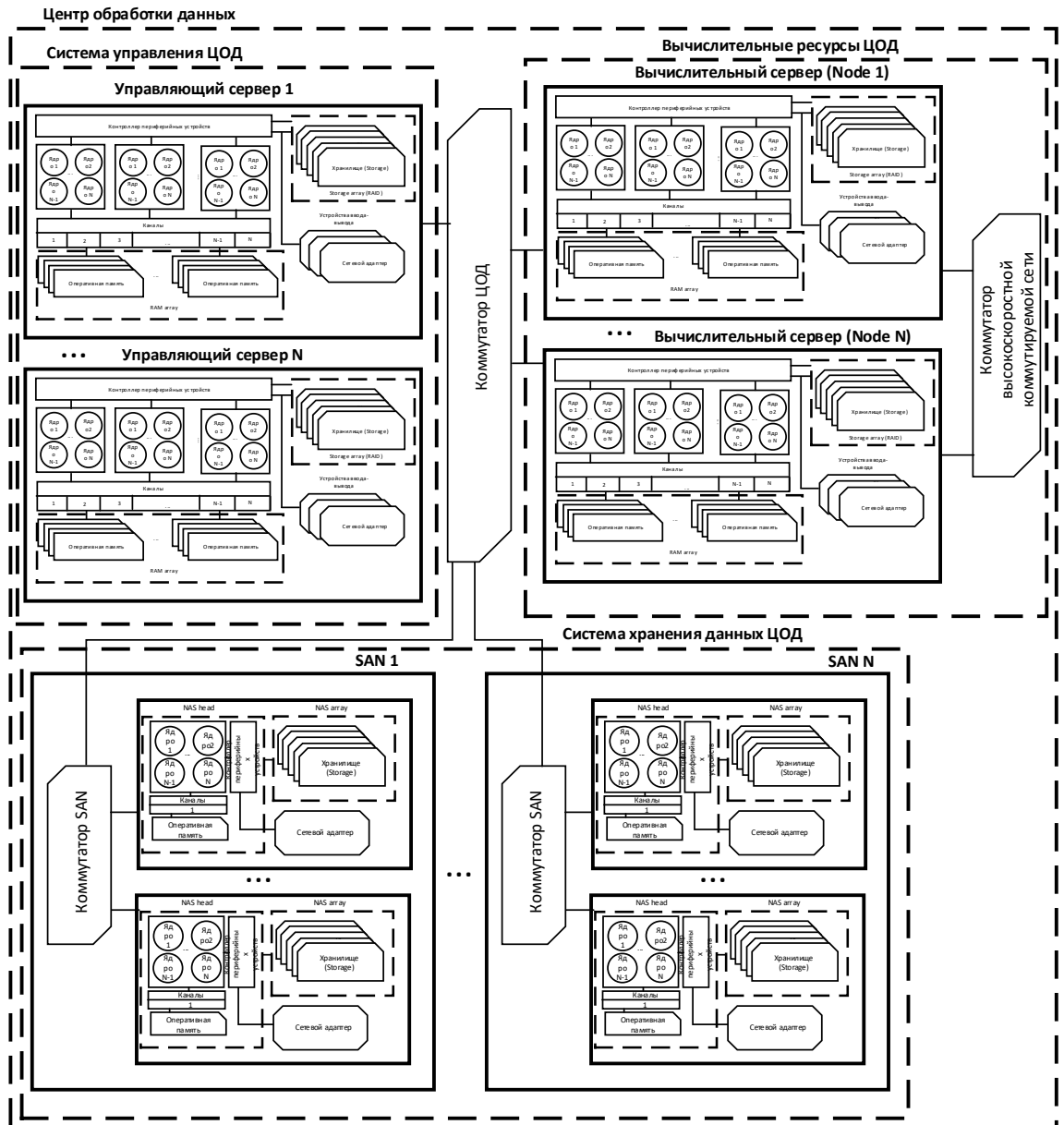


Рисунок 1.5 – Обобщенная схема центра обработки данных на основе систем виртуализации и программно-определяемых сетей

На рисунке 1.6 представлена экстраполяция статистики [10] масштабируемости IoT-систем различного класса в Российской Федерации на период с 2018 по 2025 годы с учетом роста их терминальной базы и функционального назначения IoT-терминалов.

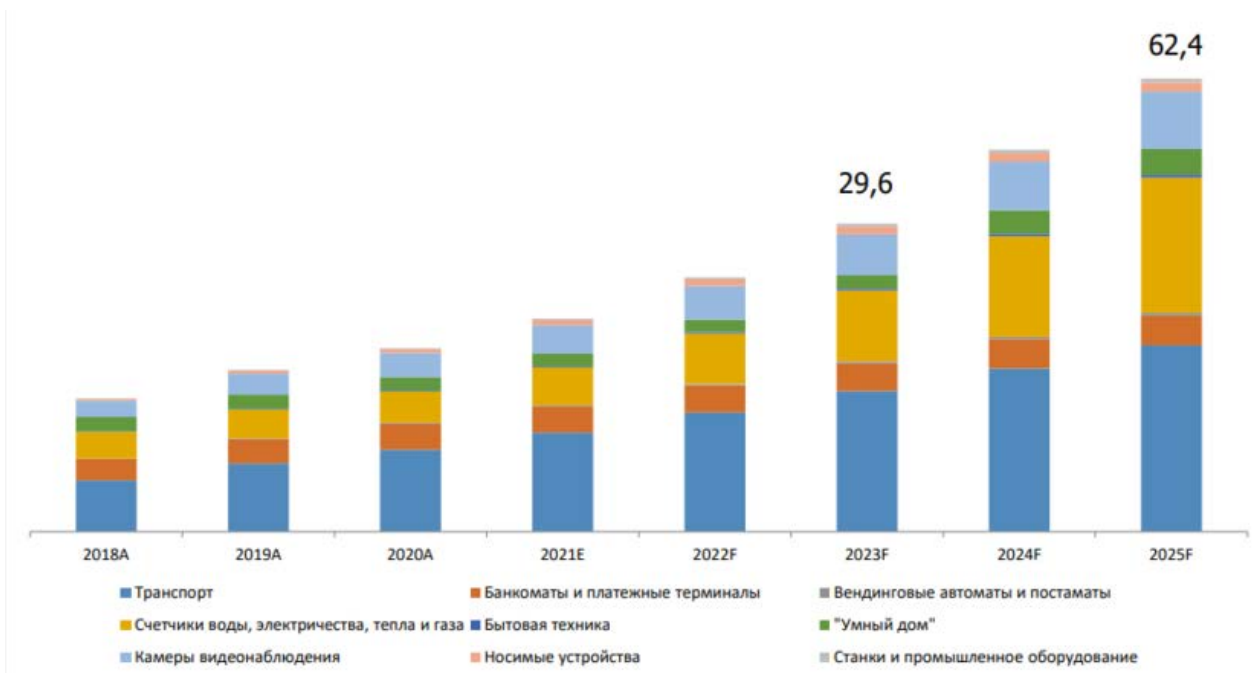


Рисунок 1.6 – Статистика роста терминальной базы IoT-систем, функционирующих в Российской Федерации (представлен в миллиардах единиц)

Аналогичная ситуация наблюдается в предметной области управления транспортно-дорожным комплексом [11, 12]. На рисунке 1.7 представлен прогноз роста рынка автотранспорта, являющийся прямым фактором возникновения проблем управления ТДК, основанных на фиксированной (или недостаточно оперативно развиваемой) пропускной способности дорожного комплекса.

Представленный на рисунках 1.6 и 1.7 рост терминальной базы IoT-систем параллельно порождает проблему роста трафика данных, передаваемых в их рамках. На рисунке 1.8 обобщена статистика и выполнена экстраполяция роста трафика, порождаемого IoT-системами в IP-сетях на уровне мультисервисных систем провайдера (рисунок 1.1) [13]. Из рисунка следует, что за период с 2015 по 2025 прогнозируется шестикратное увеличение трафика данных, порождаемого IoT-системами, что накладывает дополнительные требования на функциональность систем управления транспортной инфраструктуры.

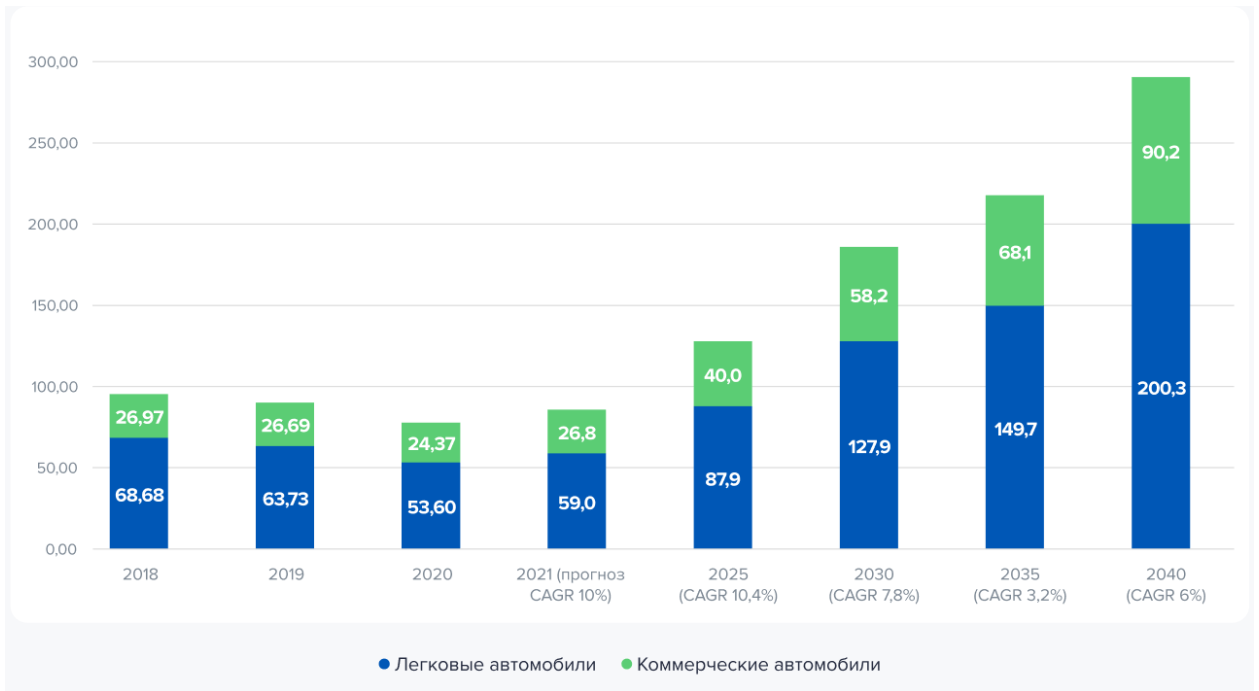


Рисунок 1.7 – Статистика мирового роста транспортной инфраструктуры

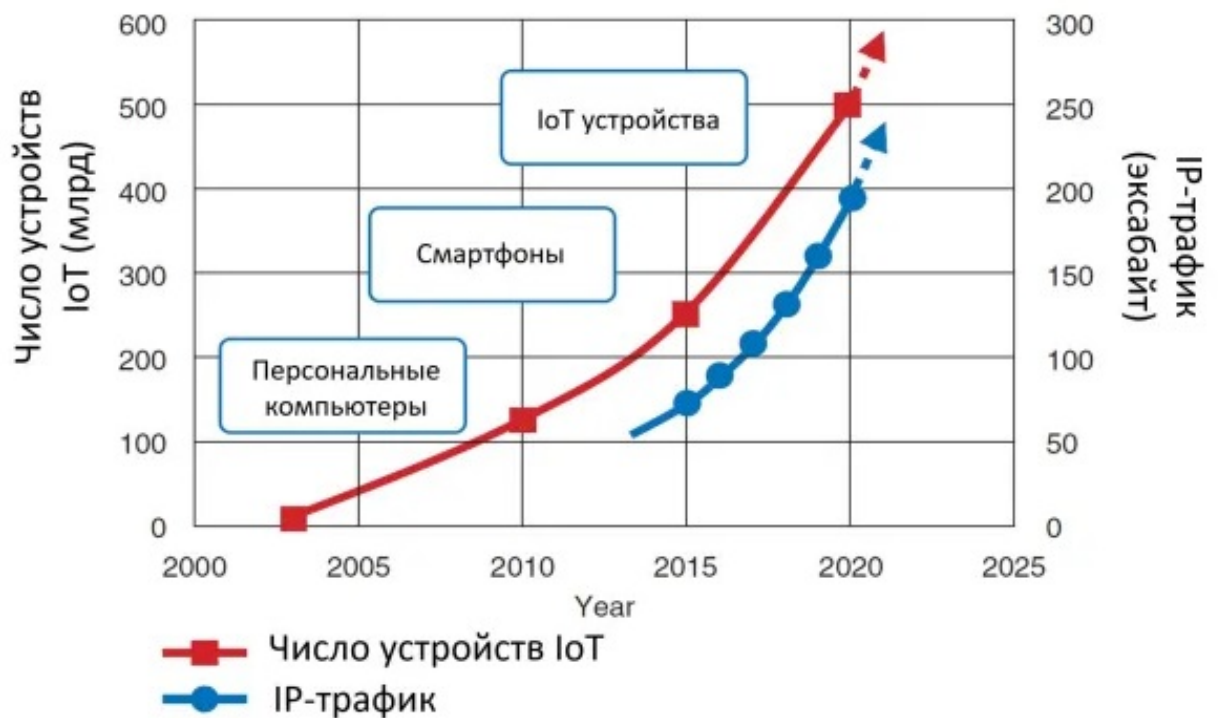


Рисунок 1.8 – Статистика мирового роста трафика данных, порождаемого IoT-системами

Очевидно, что возрастающий объем передаваемых данных опосредованно влияет на уровни облачной инфраструктуры и инфраструктуры ЦОД. Тенденцией

поддержания высокой доступности вычислительных ресурсов этих уровней является их виртуализация. Обобщение статистики по использованию виртуализированных ресурсов в рамках инфраструктуры ЦОД (рисунок 1.9) показывает, что уже начиная с 2009 года число виртуализированных ЦОД превысило число физически-организованных ЦОД (концепция on-premise) и показывает существенный ежегодный прирост [14].

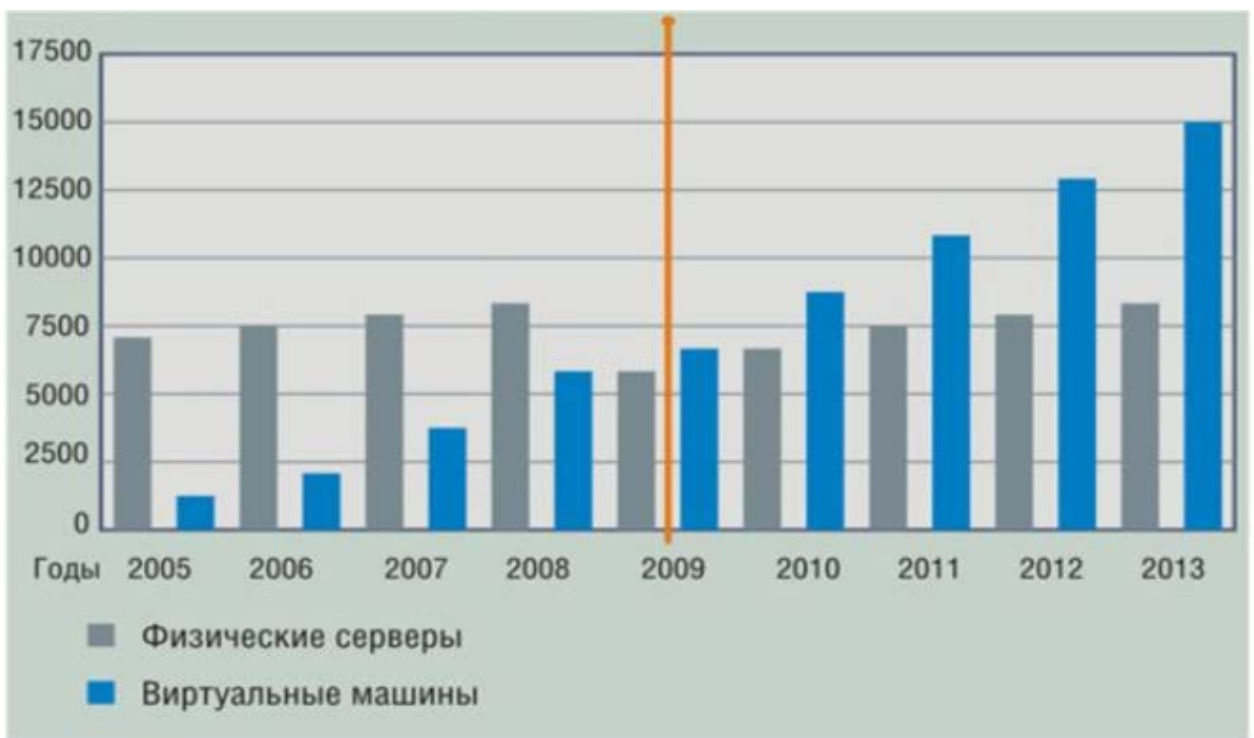


Рисунок 1.9 – Статистика прироста числа виртуализированных ЦОД (до 2013 года включительно)

Представленные выше примеры распределенных вычислительных систем в многоуровневой инфраструктуре обобщенной IoT-системы являются наиболее показательными с точки зрения вопросов потенциального роста нагрузки и связанной с ней необходимости масштабирования их информационной и вычислительной мощностей.

Безусловно, существует достаточно большое число подобных систем для других предметных областей, которые при этом также попадают в определенный выше класс высоконагруженных распределенных систем.

Очевидно, что указанные архитектурные особенности требуют развития и совершенствования систем управления подобными распределенными вычислительными системами, для которых последние являются объектами управления (ОУ).

1.2 Архитектура систем мониторинга распределенных вычислительных систем и проблематика нестационарности нагрузки на них

Неотъемлемой частью архитектуры систем управления в распределенных вычислительных системах является система мониторинга (СМ) состояния этих систем в целом и их отдельных узлов.

В общем случае под мониторингом понимается совокупность функций постоянного и/или периодического наблюдения за процессами, протекающими на объекте мониторинга (ОМ) и изменяющимися под воздействием некоторых внутренних и внешних факторов. Результаты таких наблюдений используются для формирования обоснованного управленческого воздействия на ОМ или его отдельные составляющие – узлы мониторинга (УМ). В [15, 16] рассматривается роль и место СМ (рисунок 1.10), а также выделяются три основные функции СМ, получившие развитие в рамках теории мониторинга: наблюдение, прогноз, управление.

Обобщение результатов информационного поиска в предметной области СМ для высоконагруженных распределенных вычислительных систем различного класса позволило определить их базовую архитектуру, которая является инвариантной относительно уровней распределенной вычислительной системы (рисунок 1.1). В общем виде эта базовая архитектура представлена на рисунке 1.11.

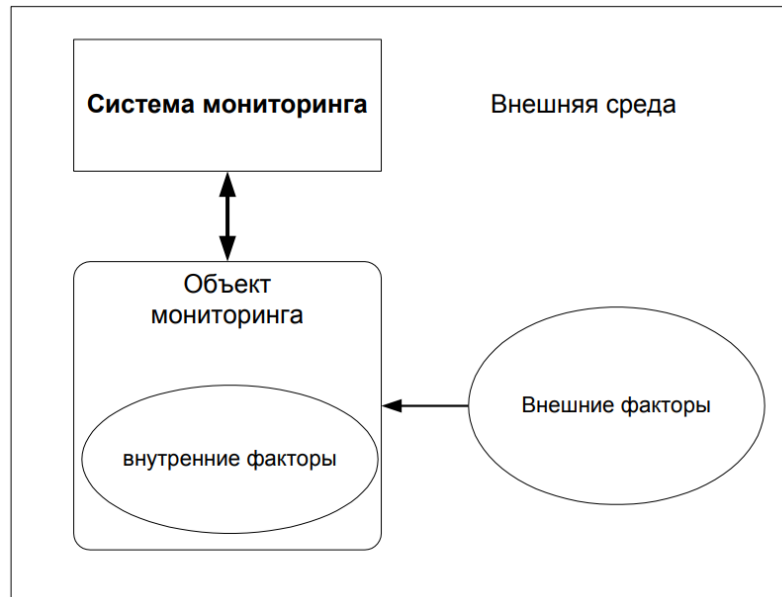


Рисунок 1.10 – Обобщенное представление места системы мониторинга

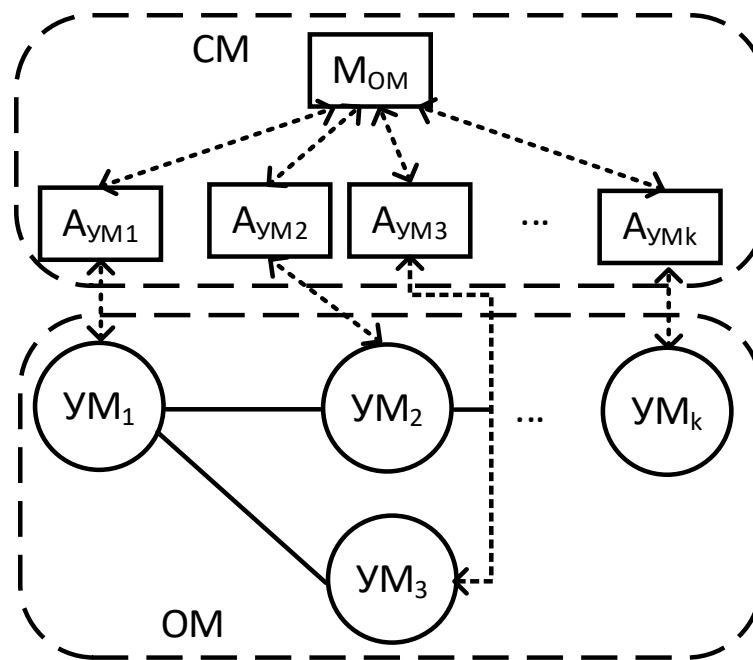


Рисунок 1.11 – Базовая архитектура системы мониторинга

Из рисунка 1.11 видно, что в общем случае для распределенного ОМ, представленного множеством $\{УМ_1, УМ_2, \dots, УМ_k\}$, СМ базируется на схеме «агент-менеджер» [17], обобщенное представление которой дается на рисунке 1.12.

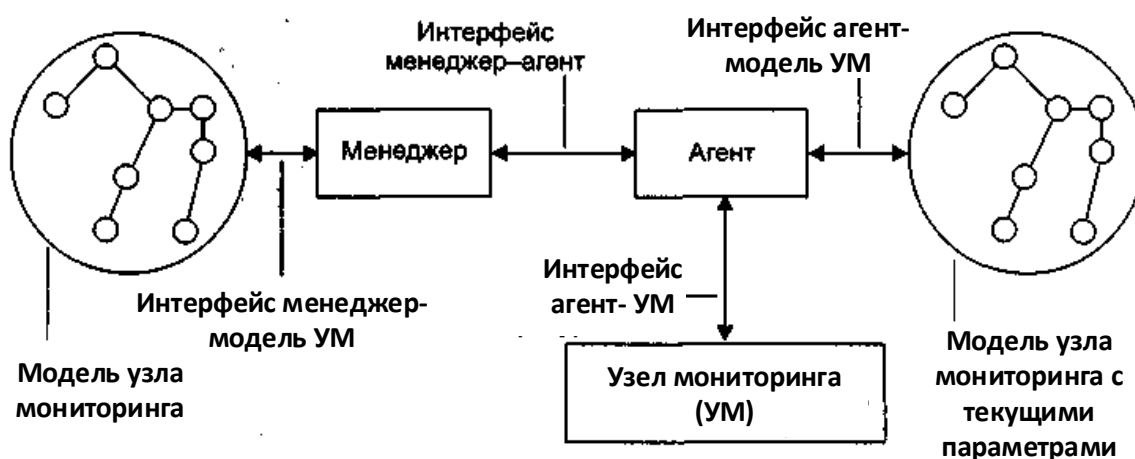


Рисунок 1.12 – Обобщенное представление схемы управления «агент-менеджер»

Из рисунка 1.12 следует, что агент – это специализированный программный или программно-аппаратный узел СМ, связанный одновременно с УМ и менеджером – специализированным узлом УМ с другой функциональностью. Для управления УМ на узле-менеджере формируется модель УМ, отражающая его наиболее значимые с точки зрения управления параметры. Подобная модель УМ формируется и на узле-агенте и отражает текущие значения выбранных параметров УМ. Узел-агент с помощью специализированного интерфейса и/или протокола передает узлу-менеджеру нормализованные значения текущих параметров УМ. Узел-менеджер обобщает полученные значения и формирует управляющее воздействие, которое передается либо узлу-агенту, либо напрямую УМ. Периодически повторяющаяся последовательность указанных выше действий узлов агента и менеджера именуется *циклом мониторинга*. Период времени между передачей текущих значений параметров именуется *скважностью цикла мониторинга*.

Таким образом, представленная на рисунке 1.11 структура СМ носит централизованный характер, функцию центра при этом выполняет узел-менеджер M_{OM} , ведомыми узлами которого является множество узлов-агентов

$$\{A_{UM1}, A_{UM2}, \dots, A_{UMk}\}.$$

Показательным примером СМ с централизованным управлением является интегрированная система управления сетью связи (Integrated network management system, INMS [18, 19], формируемая в различных конфигурациях МСС и других видов инфокоммуникационных систем с целью оценки и управления качеством предоставляемых инфокоммуникационных услуг на основе соглашения об уровне предоставления услуги (англ. Service Level Agreement, SLA) [18]. В обобщенном виде структура INMS представлена на рисунке 1.13.

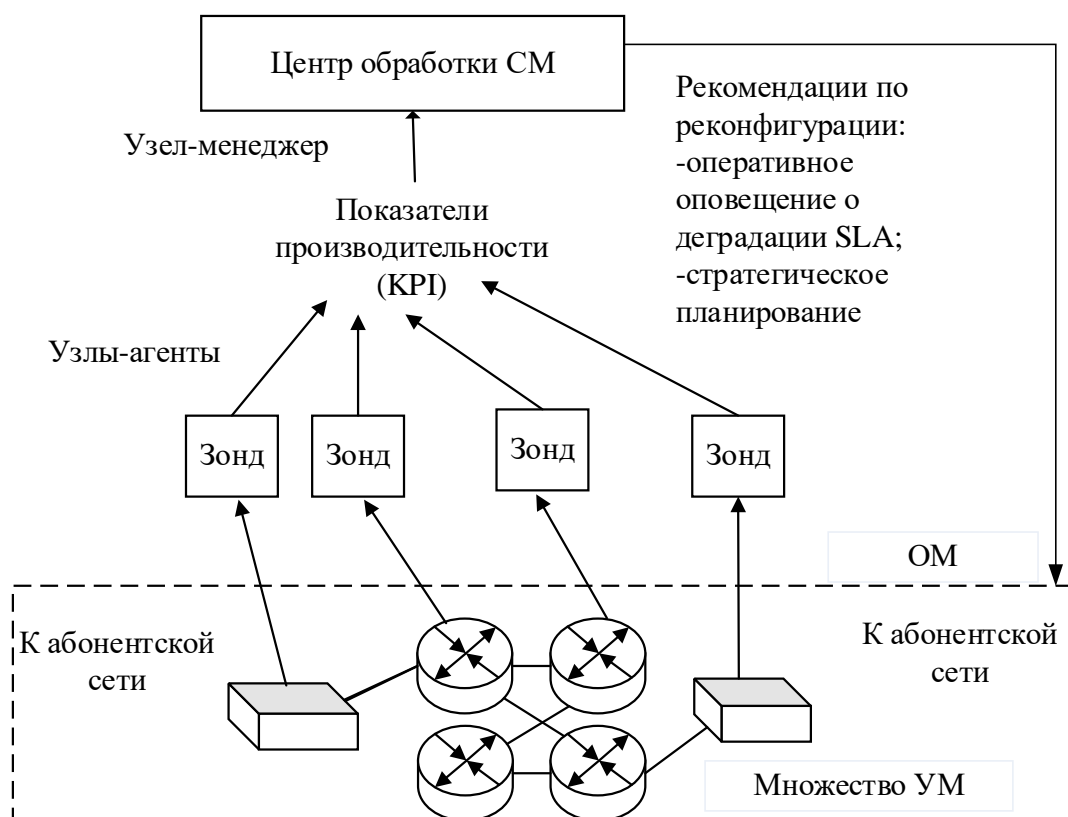


Рисунок 1.13 – Обобщенная схема организации системы мониторинга INMS

Из рисунка 1.13 видно, что INMS состоит из:

- центра обработки (узел-менеджер);
- множества зондов (узлы-агенты).

В качестве зондов выступают пассивные устройства, которые собирают ключевые показатели эффективности (Key Performance Indicator - KPI) УМ и передают их в центр обработки регулярно или по запросу, а центр обработки

выступает в виде активной структуры, которая анализирует множество KPI во времени, собирает статистику и на ее основе выстраивает профили реконфигурации сетевых устройств и рассылает оперативное оповещение о деградации KPI.

В качестве KPI в большинстве подобных систем выступают:

1. Полоса пропускания (англ. *Bandwidth*). Описывает номинальную пропускную способность среды передачи информации, определяет ширину канала.

2. Задержка доставки пакета (*IPTD*, англ. *IP packet transfer delay*) – задержка при передаче пакета, измеряется в миллисекундах. Время доставки пакета между источником и получателем для всех пакетов – как успешно переданных, так и пораженных ошибками.

3. Колебание (вариация) задержки при передаче пакетов (*IPDV*, англ. *IP packet delay variation*) – джиттер, проявляемый в том, что последовательные пакеты прибывают к получателю в нерегулярные моменты времени.

4. Потери пакетов (*IPLR*, англ. *IP packet loss ratio*) – количество (вероятность) пакетов, отбрасываемых сетью во время передачи.

Примером реализации INMS с подобной структурой является CM wiSLA (well integrated SLA) – интегрированная платформа для мониторинга и управления качеством услуг связи, IT-инфраструктуры и облачных сервисов [19]. Место включения узлов-агентов CM wiSLA в структуру логического канала типовой МСС представлена на рисунке 1.14.

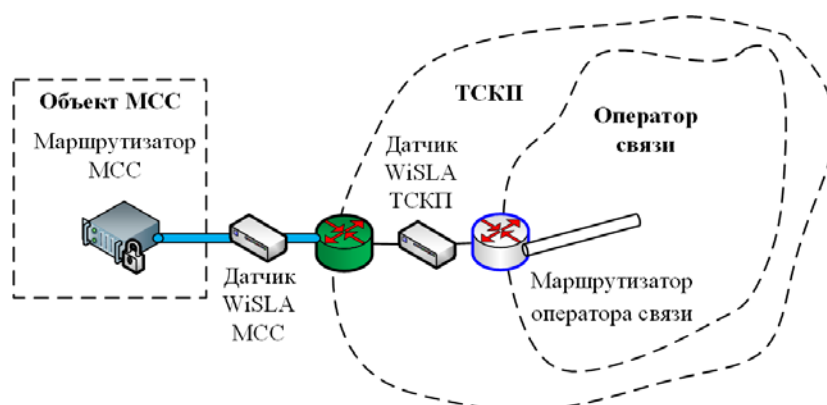


Рисунок 1.14 – Место узлов-агентов CM wiSLA в структуре МСС

Еще одним вариантом СМ централизованного типа с многоуровневой (иерархической) структурой схемы «агент-менеджер» является система управления центром обработки данных (Data Center Infrastructure Management, DCIM), вариант которой (рисунок 1.15), обеспечивающий мониторинг ресурса памяти виртуализированных узлов обработки данных, представлен в [20].

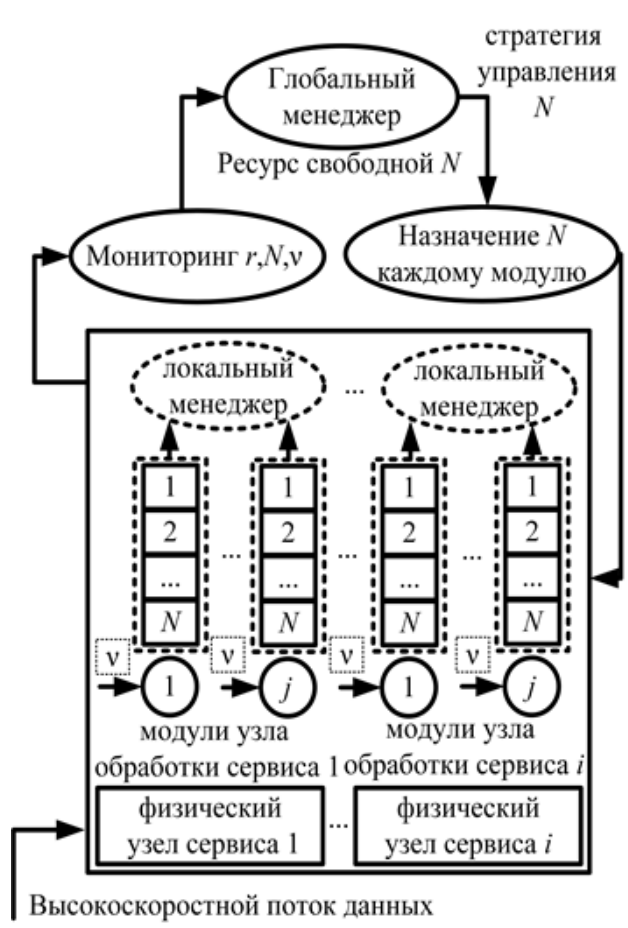


Рисунок 1.15 – Многоуровневая структура СМ в распределенной вычислительной системе обработки высокоскоростных потоков данных

Примером реализации подобной СМ в реально действующих ЦОД является СМ Zabbix [21], схема программных модулей которой представлена на рисунке 1.16.

Практика эксплуатации подобных систем, а также проводимые исследования их эффективности для различных условий функционирования ОМ [20, 23], для которых они разрабатывались, показывает, что реализация их целевой функции

чаще всего выполняется в стационарных (штатных условиях эксплуатации, при которых отклонения значений КРІ, фиксируемых узлами-агентами соответствуют некоторому известному закону распределения.

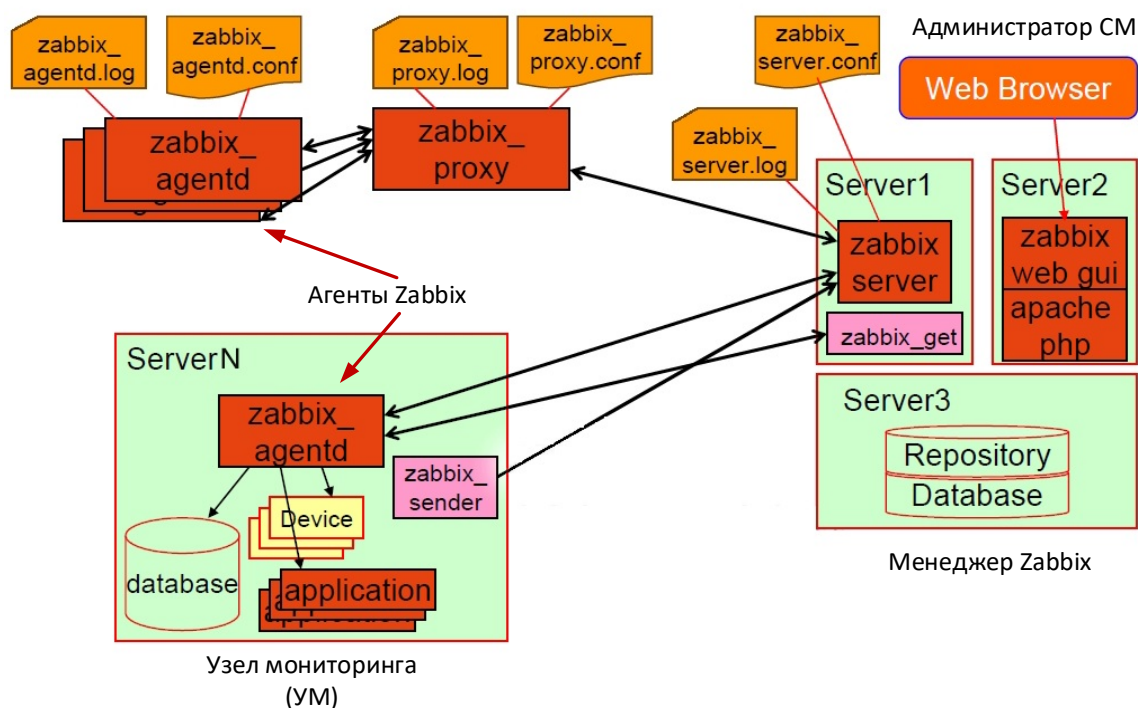


Рисунок 1.16 – Структура программных модулей системы мониторинга Zabbix

В общем случае недостатками подобных СМ являются [16]:

- структурная инертность, связанная с достаточно длительной скважностью цикла мониторинга;
- отсутствие реализации механизмов реагирования на незначительные изменения КРІ на длительных временных периодах функционирования ОМ и возможность управления только в случаях «взрывного» характера значений КРІ на коротких временных периодах.

При этом совершенствование функциональности систем, являющихся ОМ для рассматриваемых видов СМ в настоящее время приводит к ряду проблем, требующих исследования и разработки нового математического и программного обеспечения для их решения.

Одной из таких проблем является возникновение на крупномасштабных многоузловых ОМ нестационарной нагрузки, имеющей временной лаг распространения по узлам ОМ.

В общем случае в предметной области систем управления под *временным лагом* (time lag или lag time) понимается смещение уровней временного ряда значений оцениваемых параметров относительно некоторого заданного значения в положительную (положительный лаг, лаг запаздывания) или отрицательную (отрицательный лаг, лаг опережения) [22].

Пример представления временного лага для системы управления водосбросом в условиях ливневых осадков представлен на рисунке 1.17.

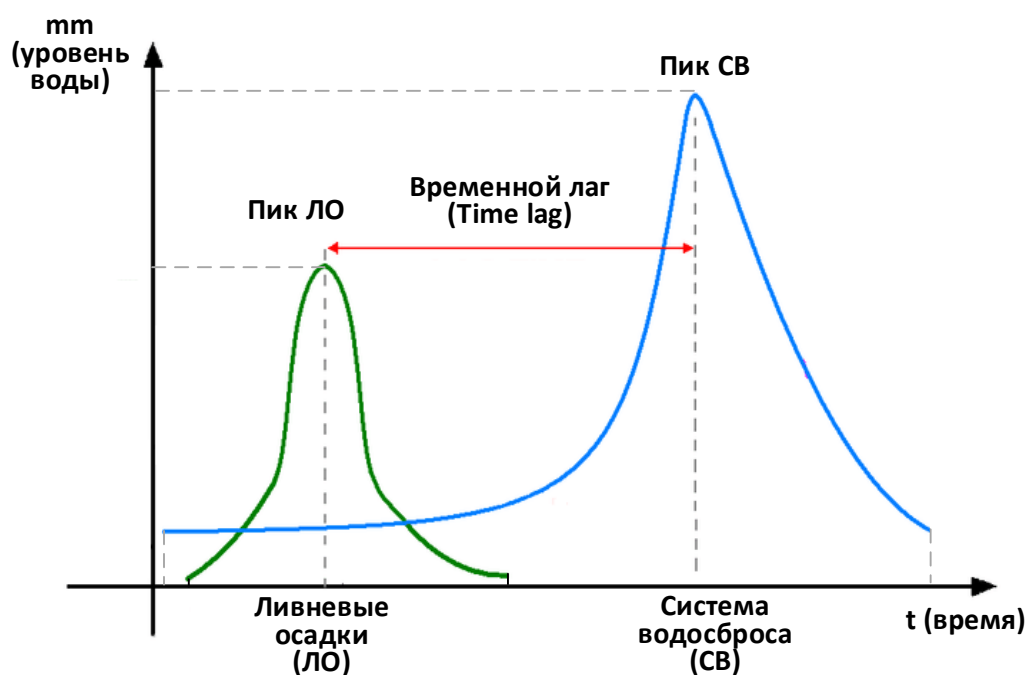


Рисунок 1.17 – Представление временного лага на примере системы управления водосбросом

Из рисунка видно, что пиковое значение параметра «уровень» воды» в системе управления водосбросом запаздывает относительно пикового значения «уровень воды» ливневых осадков (ЛО), которые для системы водосброса (СВ) выполняют роль нагрузки. Разность значений временного ряда пиковых значений

этого параметра и представляет из себя значение временного лага. По-другому его можно назвать скоростью реакции системы на изменение некоторого параметра.

В СМ под временным лагом или лагом управления (control lag) [24, 25] распространения нагрузки на УМ понимается смещение уровней временного ряда значений параметров нагрузки относительно первоначального положения на некоторое значение h моментов времени (рисунок 1.18).

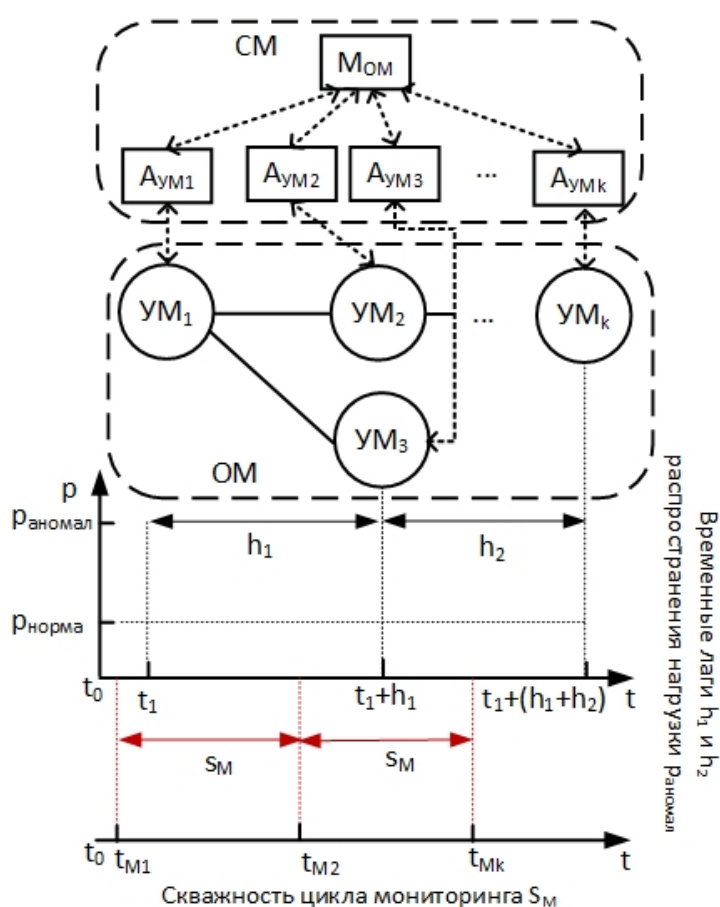


Рисунок 1.18 – Представление проблемы нестационарности нагрузки, имеющей временной лаг распространения по узлам мониторинга

Например, увеличение в момент времени t_1 на узле доступа МСС количества терминальных устройств, порождающих исходящий трафик, приведет к переполнению буфера пакетов отдельных направлений в узлах управления сообщениями в моменты времени с некоторым значением t_1+h , что потребует формирования на них управляющего воздействия по модификации маршрутной

информации, балансирующей аномальность такого трафика. Таким образом, имеет место положительный временной лаг (запаздывание) реакции ОМ на управляющее воздействие. Аналогичная проблема имеет место и на ОМ других предметных областей (управление транспортными потоками, балансировка виртуализированных ресурсов в центрах обработки данных и т.д.). Этот временной промежуток может быть довольно значительным, особенно при работе с большими объемами данных и существенно зависеть от:

- физического и/или логического расстояния между контролируемыми УМ;
- латентности протокола взаимодействия узлов-агентов с узлом-менеджером.

Показательным примером многоузлового ОМ, в рамках которого возникающая нагрузка может иметь временной лаг распространения, является реализация многоуровневой IoT-системы (рисунок 1.1) на основе стандарта NB-IoT [26] (Narrow Band Internet of Things), получающего все более широкое распространения в M2M (Machine-To-Machine) вариантах IoT-систем.

Проблема возникновения временного лага распространения нагрузки в подобных системах является комплексной и зависит от:

- цикла рабочего состояния NB-IoT;
- периодов пиковой и обычной нагрузки, определенных в рекомендациях ITU-T E.492 и ITU-T E.500 [27, 28].

На рисунке 1.19 представлена обобщенная временная диаграмма цикла рабочего состояния NB-IoT.

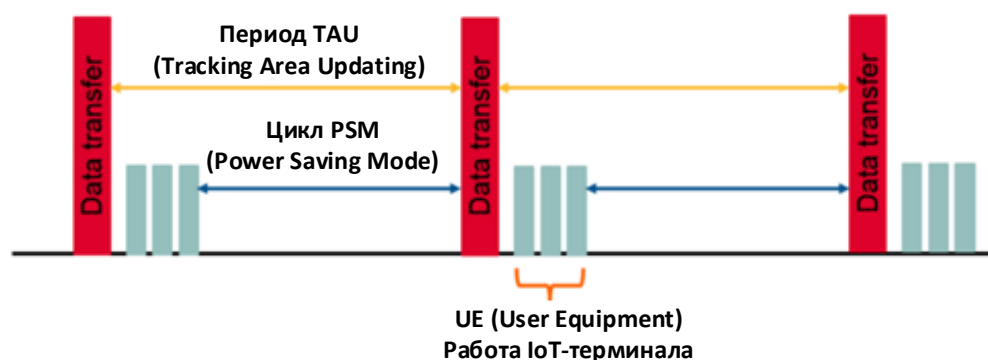


Рисунок 1.19 – Диаграмма цикла рабочего состояния NB-IoT

Из рисунка 1.19 видно, что на длительных временных периодах цикл рабочего состояния NB-IoT представлен периодами TAU (Tracking Period Updating), которые в свою очередь состоят из:

- периода UE (User Equipment) - активной работы IoT-терминала;
- цикла энергосберегающего простоя PSM (Power Saving Mode).

Периоды UE и TAU определяются таймерами T3424 и T3412 соответственно, а длительность цикла PSM определяется как разность $PSM=TAU-IDLE$ (рисунок 1.20).

Подобная цикличность активности и простоя множества IoT-терминалов, находящихся в разных периодах порождает нестационарность потока их обмена данными при подключении разных подмножеств IoT-терминалов в периоды пиковой и обычной нагрузки к уровню базовых станций и их перемещении в пространстве. Очевидным следствием этого является задержка времени «всплеска» нестационарной нагрузки на расположенных рядом базовых станциях, относительно вектора направления движения подмножества активных IoT-терминалов.

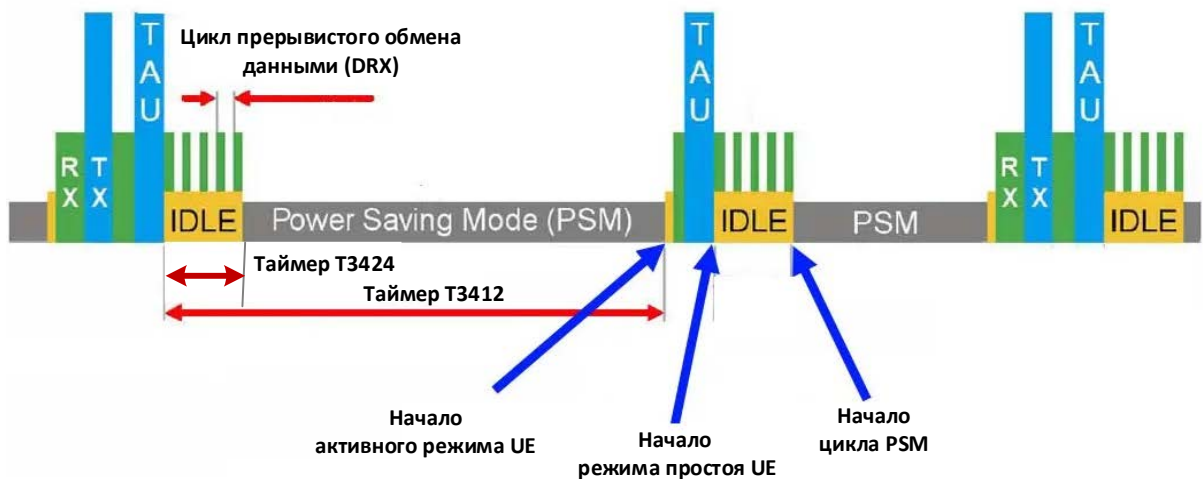


Рисунок 1.20 – Детализация таймингов цикла рабочего состояния NB-IoT

С точки зрения СМ подобной IoT-системы, для которой одним из уровней мониторинга является уровень базовых станций указанная временная задержка будет представлять временные лаги h_1 и h_2 (рисунок 1.18).

Анализ рынка подключенных к IDS-системе автомобилей (схема M2M) показывает [29], что в часы пиковой нагрузки трафик обмена данными в определенных сотах уровня базовых станций можеткратно возрастет (traffic uplift) и с некоторой задержкой h_k перемещаться от соты к соте (рисунок 1.21).

Аналогичная ситуация наблюдается в многоузловых распределенных вычислительных системах для высокоскоростной обработки данных, функционирующих на уровне ЦОД (рисунок 1.1) и рассматриваемых в [20]. Их обобщенная схема представлена на рисунке 1.22.

В подобных системах для определения степени «аномальности» потока данных, приводящей к временному лагу его распространения на отдельные узлы обработки подпотоков используются следующие параметры:

- DPP - пиковый период за день (Day Peak Period) – фиксация наибольшего объема трафика, измеренного за день. Получение этого параметра требует непрерывных измерений и используется для каналов, где час может варьироваться в разные дни;

- FDMI - фиксированный интервал измерений за день (Fixed Day Measuring Interval). Измерение этого параметра выполняется в заранее известные пиковые моменты времени. Этот метод используется, когда пиковые периоды наступают с регулярными интервалами (например, праздничные дни, часы пик и т.д.).

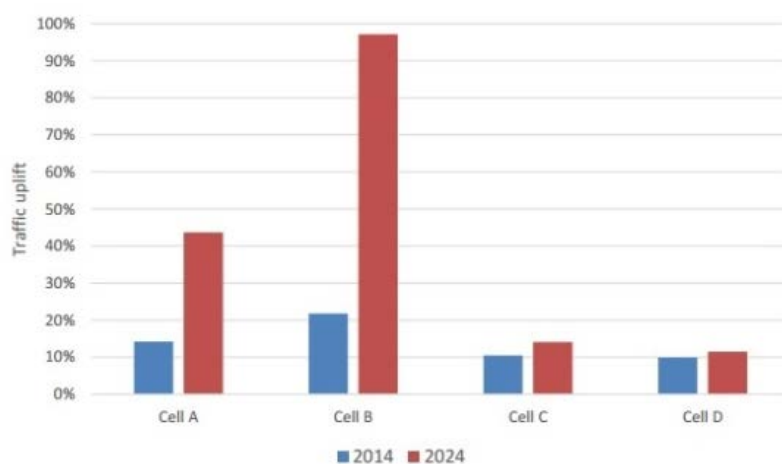


Рисунок 1.21 – Всплеск трафика в пределах соты (сота В) в часы пиковой нагрузки IDS-системы

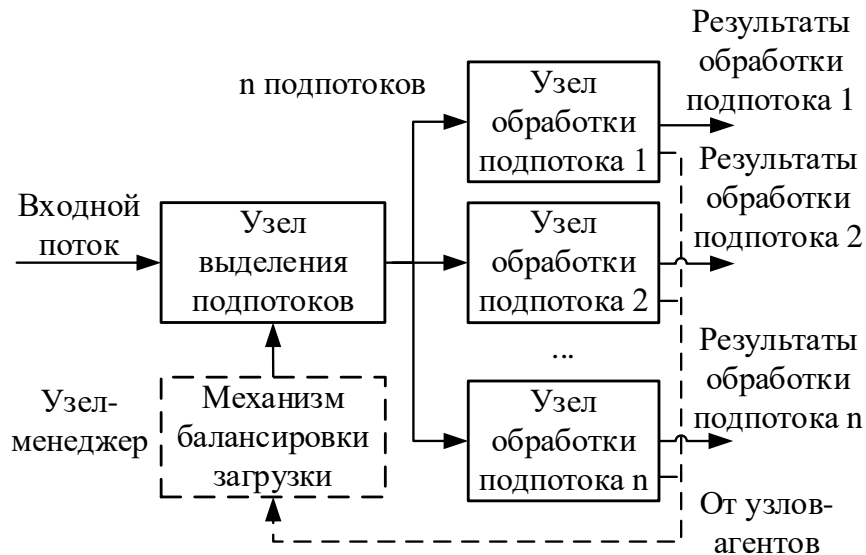


Рисунок 1.22 – Структура распределенной вычислительной системы высокоскоростной обработки данных

Пример «всплеска» нестационарности нагрузки на конкретных вычислительных модулях (ВМ) относительно ожидаемого значения интенсивности нагрузки представлен на рисунке 1.23.

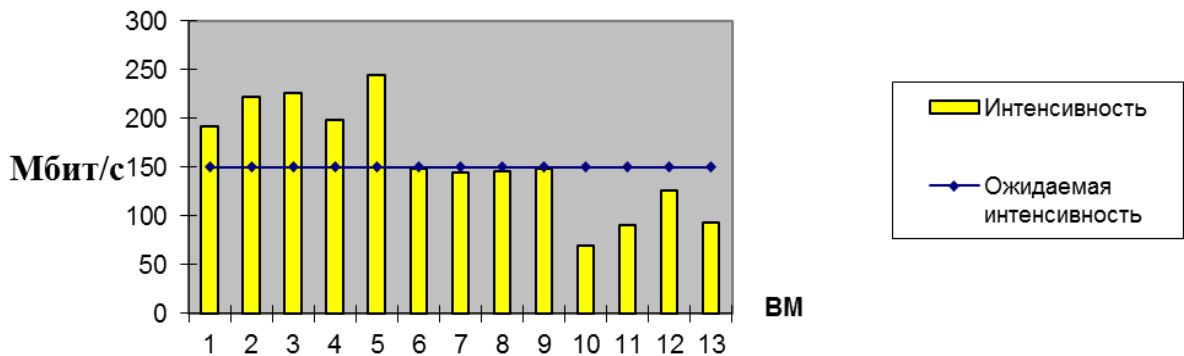


Рисунок 1.23 – Результат анализа нестационарности нагрузки на отдельных ВМ системы высокоскоростной обработки данных [20]

Таким образом для указанных случаев СМ с централизованной архитектурой не в полной мере обеспечивают учет временного лага распространения, особенно

если s_M – скважность цикла мониторинга (обмена данными между агентами и менеджером СМ) не совпадает со значениями смещения h . Чем выше значение скважности цикла, тем быстрее происходит мониторинг и тем эффективнее работает система. Однако, высокий уровень скважности цикла может привести к увеличению нагрузки на центральный узел и увеличению времени отклика системы.

Проблема несовпадения моментов скважности цикла мониторинга может привести к тому, что данные, полученные в момент, когда нагрузка уже изменилась, будут использоваться для анализа и принятия решений. Это может привести к ошибочным выводам и неправильным решениям.

1.3 Анализ методов моделирования процесса мониторинга при нестационарной нагрузке

Очевидным подходом решения представленной в п. 1.2 проблемы нестационарности нагрузки в условиях временного лага ее распространения для существующих архитектур СМ является решение задачи прогнозирования параметров этой нагрузки на УМ в направлении вектора ее распространения.

В общем виде эта задача связана с получением значений параметров нагрузки на УМ не только в момент времени t , но и в момент времени $t+k$, где k – промежуток времени такой, что $k \leq h$. Таким образом, СМ, имея прогнозные значения параметров нагрузки на момент времени, значения которого меньше значения лага распространения этой нагрузки, может упреждающе (проактивно) передать управляющее воздействие на УМ или подмножество УМ, в направлении которого распространяется вектор нагрузки.

Анализ исследований, посвященных решению указанной задачи показал, что за период развития и совершенствования математического и программного аппарата современных СМ было предложено несколько подходов, базирующихся на использовании различных технологий.

Наиболее распространенным является подход, основанный на моделировании нагрузки (например, потока передаваемых данных, транспортного трафика, потока запросов на использование виртуализированных ресурсов ЦОД) и решение задачи прогнозирования моментов нестационарности в рамках теории систем массового обслуживания, в частности на основе аппарата сетей массового обслуживания (СеМО), позволяющих реализовать модели, учитывающие нестационарность нагрузки на некотором периоде времени. Так в работах [20, 30, 31], связанных с функционированием СМ МСС, было определено, что в условиях нестационарности («всплесков») нагрузки, потоку свойственна пачечность передаваемых пакетов и случайная продолжительность, и периодичность моментов пиковой нагрузки (моментов нестационарности). Также было определено, что наличие таких моментов не позволяет описать нагрузку моделью пуассоновского потока. Для описания произвольного распределения длительности случайной величины обосновано предлагается использование смеси экспоненциальных (гиперэкспоненциальное распределение). Для моделирования лавинообразного роста нагрузки гиперэкспоненциальное распределение, представляется двух случайных величин ($r=2$), распределенных по двум экспоненциальным законам (фазам). При этом вероятность возникновения случайной величины, в i -ой фазе рассчитывается выражением 1.

$$q_i (i=\overline{1,r}), \sum_{i=1}^r q_i = 1 \quad (1)$$

Нестационарность нагрузки интервала τ и заданным коэффициентом вариации V «всплесков» нагрузки, таким образом, можно описать на основе схемы распределения H_2 (от «гиперэкспоненциальное»), представленной на рисунке 1.23.

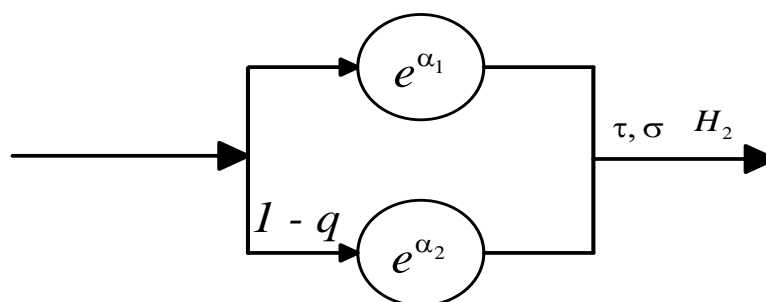


Рисунок 1.23 – Представление нестационарной нагрузки с резкими «всплесками» на основе схемы распределения H_2

Для моделирования УМ, обслуживающего нагрузку подобного вида, было предложено использование СМО $H_2/M/1/n$ при ограничении на время обслуживания единичной заявки $T_{обсл} = 1/\mu = 1$ (рисунок 1.24):

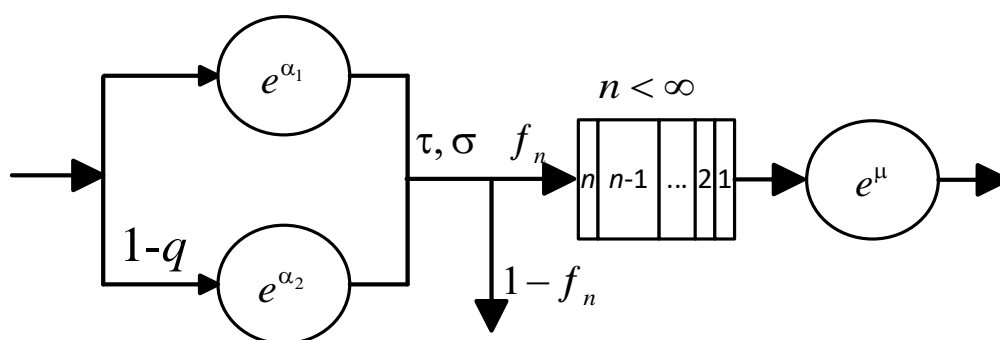


Рисунок 1.24 – СМО-модель $H_2/M/1/n$ УМ с нестационарным характером входной нагрузки

В [20] в качестве базового алгоритма функционирования узла-менеджера СМ для подобных УМ предлагается алгоритм гистерезисного управления. Схема изменения статуса S моментов нестационарности нагрузки представлена на рисунке 1.25, а схема СМО для подобного типа управления представлена на рисунке 1.26.



Рисунок 1.25 – Представление статуса моментов нестационарности (перегрузка) в гистерезисном алгоритме управления

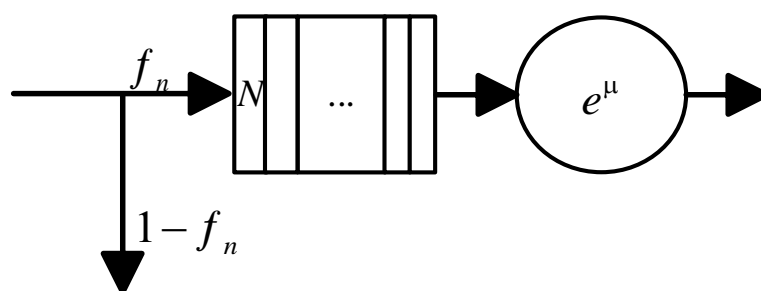


Рисунок 1.26 – Схема СМО УМ с гистерезисным управлением моментами нестационарности

Указанный подход обеспечивает достаточно высокую эффективность обнаружения и управления нестационарным характером нагрузки, однако, с точки зрения рассматриваемой в исследовании проблемы, имеет ряд ограничений:

- использование схемы распределения H_2 позволяет моделировать только лавинообразный характер нагрузки (режим перегрузки), в то время как в современных распределенных вычислительных системах нестационарность может иметь эффект «накопления» - прироста небольших значений. Использование схем

распределения с другим коэффициентом вариации ($v \geq 2$) существенно усложняет модель;

- подобный подход не позволяет рассматривать проблему временного лага распространения нагрузки, усредняя ее значения на входе УМ, вне зависимости от источника ее поступления.

Кроме того, представленный выше подход к моделированию процесса функционирования УМ на основе положений теории массового обслуживания требует априорного знания параметров всех УМ в составе ОМ, что, как правило, на практике является мало реализуемым.

В [32, 33, 34] для описания подмножества взаимосвязанных УМ, как стохастической динамической системы, предлагается использование статистических методов моделирования, с помощью ее представления в виде уравнений состояния и наблюдения. В указанных работах моделирование узлов МСС (вариант УМ) выполняется на основе модели фильтров Калмана (линейная динамическая система) и скрытых марковских моделей (Hidden Markov Model, НММ). Так фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной стохастической динамической системы. Расчёт текущего состояния такой системы опирается на текущие измерения ее параметров, а также предыдущее состояние фильтра Калмана. Алгоритм фильтрации является двухэтапным:

1. Этап прогнозирования – экстраполяция значений переменных состояния, а также их неопределённости.

2. Уточнение результата экстраполяции, полученного на этапе 1, по данным текущих измерений параметров.

То есть, модель на основе фильтра Калмана потенциально позволяет идентифицировать состояние подмножества УМ, опираясь на текущие измерения и информацию о его предыдущем состоянии.

В общем виде она может быть представлена выражением 2:

$$\begin{aligned} X_t &= F_t X_{t-1} + B_t U_t + V_t, \\ Y_t &= H_t X_t + W_t; \end{aligned} \quad (2)$$

где: X_t – скрытый процесс (изменения состояний функционирования подмножества УМ);

Y_t – наблюдаемый процесс (изменения значений показателей, характеризующих состояние подмножества УМ);

F_t – матрица эволюции процесса X_t ;

B_t – матрица управления;

U_t – управляющее воздействие;

V_t – шум возбуждения (случайный процесс);

H_t – матрица наблюдений (измерений);

W_t – шум наблюдения (случайный процесс).

Результатом выполнения алгоритма Калмана является определение оптимальной матрицы коэффициентов K_t и ковариационной матрицы $P_{t|t}$ оценки процесса X_t (выражение 3):

$$\begin{aligned} K_t &= P_{t|t-1} H_t^T S_t^{-1}, \\ P_{t|t} &= (I - K_t H_t) P_{t|t-1}; \end{aligned} \quad (3)$$

где: $S_t = H_t P_{t|t-1} H_t^T + R_t$ – ковариационная матрица вектора отклонения (вектора ошибки);

R_t – ковариационная матрица шума наблюдения W_t ;

$(\cdot)^T$ – оператор транспонирования вектора (матрицы).

Применимость модели на основе фильтра Калмана существенно ограничивается условиями параметрической неопределенности законов распределения, характеризующих матрицы F_t и H_t (выражение 1). В работе [34] предложен способ идентификации параметров при наличии только информации о

процессе Y_t . Этот способ базируется на теории скрытых марковских моделей (СММ), в частности на рекурсивных СММ [35, 36].

Принцип функционирования рекурсивной СММ основан на оценке параметров для текущего момента времени и их перерасчете на основании данных, полученных в предыдущий момент времени. Сложностью применения рекурсивных СММ является нелинейный характер зависимостей процессов X_t и Y_t , требующий их приведения к индикаторному виду, что не всегда возможно.

Как было указано выше, рассмотренные подходы в общем случае применимы для условий параметрической определенности состояний УМ. Между тем рост масштабов распределенных вычислительных систем, усложнение их структурных и функциональных схем существенно усложняют идентификацию состояния их компонентов (УМ, с точки зрения СМ), в частности априорное представление законов распределения значений параметров нагрузки, обрабатываемых отдельными УМ или подмножествами УМ. Ряд исследований [37, 38, 39] направлены на преодоление подобного противоречия за счет описания распределения их вероятностей смесью конечного числа нормальных распределений. Использование такого подхода обусловлено наличием во временных рядах оцениваемых параметров участков локальной стационарности. Однако и подобный подход имеет существенное ограничение, связанное с упрощением представления анализируемых процессов нестационарности нагрузки в виде *мартингала* – искусственного случайного процесса, для которого наилучшим предсказанием поведения в будущем является его текущее состояние [40].

Другой немаловажной проблемой, ограничивающей применение указанных подходов, является лавинообразный рост обрабатываемых СМ данных, которые к тому же могут иметь неструктурированный характер. Этот рост обусловлен как развитием возможностей сенсоров, поддерживаемых узлами-агентами, закрепленными за УМ, так и ростом числа подобных сенсоров [41]. Так текущие реализации беспилотных автомобилей (*self-driving vehicle*) имеют десятки

сенсоров, необходимых для функционирования их вычислительной системы (рисунок 1.27).

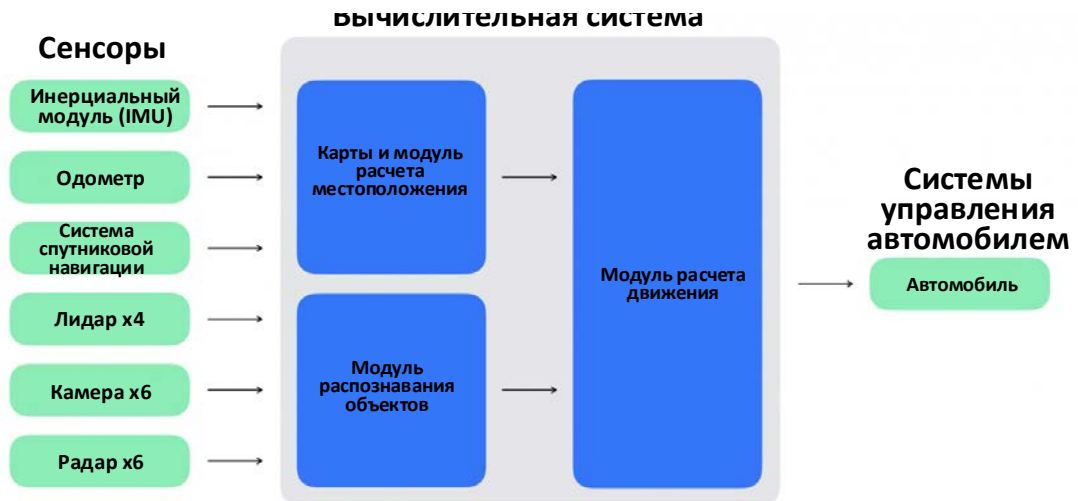


Рисунок 1.27 – Структурная схема компонентов беспилотного автомобиля

Подобные тенденции наблюдаются и для иных крупномасштабных ОМ, таких, например, как системы «умный дом», «умный город». В системах МСС, обеспечивающих инфокоммуникационный слой таких ОМ, подобные тенденции связаны с ростом числа мобильных терминалов, также поддерживающих множество сенсоров и формирующих в пространственно-временной области многомодальный трафик данных.

Учет этих тенденций отразился на исследованиях в области решения задачи распознавания моментов нестационарности нагрузки с использованием интеллектуальных вычислительных подходов, таких как методы нечеткой логики, эволюционные алгоритмы, байесовские и нейронные сети. Так в исследовании [42] развернутый анализ применения глубоких сетей доверия (deep belief network, DBN) в вопросах прогнозирования аномальности автомобильного трафика, и предложена структура глубокой сети доверия с многозадачным обучением, в которой слой многозадачных регрессионных моделей используется для аппроксимации значений неконтролируемых параметров. В работах [43, 44] для прогнозирования нестационарности нагрузки предлагается ее моделирование с помощью

многоуровневого автокодировщика (Stacked Autoencoder), а в работе [45] рассматривается применение рекуррентных нейронных сетей (RNN) для моделирования моментов транспортных заторов.

Основной проблемой рассмотренных выше интеллектуальных методов является отсутствие учета пространственно-временной корреляции моментов нестационарности нагрузки, не позволяющей учесть временные лаги ее распространения между УМ, в особенности на длительных временных периодах мониторинга.

Одним из наиболее перспективных подходов, обеспечивающих вариант решения указанной проблемы, является использование разновидности рекуррентных нейронных сетей – сетей долгой краткосрочной памяти (long short-term memory, LSTM), впервые рассмотренных в [46]. Так в работе [47] для моделирования нелинейной динамики трафика была предложена трехслойная LSTM-сеть, скрытый слой которой представлял собой каскад блоков памяти, поддерживающих расчет величины временных лагов момента нестационарности трафика. При этом указанный подход ориентирован на отложенный анализ нагрузки, использующий предварительно полученные данные. Безусловно, подобное решение является приемлемым для ряда СМ с централизованной архитектурой, однако малоприменим в условиях необходимости решения задачи управления параметрами УМ в реальном времени.

Таким образом, существует объективное противоречие между необходимостью своевременной реакции СМ распределенных вычислительных систем на моменты нестационарности нагрузки на УМ, которая имеет временной лаг распространения по подмножеству УМ и недостаточной функциональностью математического и программного обеспечения СМ, обеспечивающего решение этой проблемы.

1.4 Формальная постановка задачи исследования

Исходя из проведенного анализа и цели исследования необходимо разработать математическое обеспечение процесса функционирования распределённой системы мониторинга при нестационарной нагрузке на объекте мониторинга, а также структуру программного обеспечения ее логических компонентов.

Формальная постановка задачи имеет следующий вид.

Дано:

$OM = \{UM_1, \dots, UM_k\}$ – объект мониторинга, представленный множеством узлов мониторинга;

$\bar{X}_k(t)$ – вектор параметров $(x_1, x_2, x_3, \dots, x_n)$ нагрузки k-го УМ в момент времени t ;

$CM = \{MM_{OM}, \{AM_{UM1}, \dots, AM_{UMk}\}\}$ – система мониторинга централизованного типа, представленная множеством элементов: узлом-менеджером мониторинга MM_{OM} и подмножеством узлов-агентов мониторинга AM_{UMk} .

$R = MM_{OM} \times \{AM_{UM1}, \dots, AM_{UMk}\}$ – k-местное отношение элементов множества CM (связь «многие к одному»).

S_k – скважность моментов мониторинга k-го УМ;

$T_k = \{S_k^1, \dots, S_k^j\}$ – период мониторинга k-го УМ, представленный дискретным временем скважности мониторинга S_M^j ;

$S_k^j = t_{\text{delay}}^j = t_{\text{сб}}^j + t_{\text{пр}}^j + t_{\text{обр}}^j + t_{\text{пер}}^j + t_{\text{конф}}^j$ – j-й момент скважности мониторинга:

время задержки t_{delay}^j управляющего воздействия на k-й УМ, представленное суммой времен:

$t_{\text{сб}}^j$ – время сбора параметров узлом-агентом;

$t_{\text{пр}}^j$ – время приема параметров узлом-менеджером;

$t_{\text{обр}}^j$ – время обработки параметров и формирования управляющего воздействия узлом-менеджером;

$t_{\text{пер}}^j$ – время передачи управляющего воздействия узлу-агенту;

$t_{\text{конф}}^j$ – время конфигурирования k-го УМ узлом-агентом.

$\bar{X}_k(t_{\text{сб}}^j)$ – вектор параметров $(x_1, x_2, x_3, \dots, x_n)$ нагрузки k-го УМ в момент времени $t_{\text{сб}}^j$;

$G(OM, E), \langle \{UM_i, UM_k\}, \prec \rangle \in E, UM \in OM$ – взвешенный оргграф

распространения вектора параметров нагрузки с отображениями $init: E \rightarrow OM$ и $ter: E \rightarrow OM$.

$h_{(k-1, k)}$ – временной лаг распространения вектора параметров нагрузки: вес дуги $\{UM_{k-1}, UM_k\}$ оргграфа $G(OM, E)$.

Требуется:

1. Разработать модель СМ для заданного ОМ, обеспечивающую решение задачи прогнозирования $\bar{X}_k(t_{\text{сб}}^{j+1})$ на заданном UM_k с учетом его связности в оргграфе $G(OM, E), \langle \{UM_{k-1}, UM_k\}, \prec \rangle \in E$:

2. Разработать алгоритм $\text{Alg}_{\text{AM}_{\text{UM}_k}}$ функционирования AM_{UM_k} ,

минимизирующий t_{delay}^j :

$$t_{\text{delay}}^{j+1} = f\left(\bar{X}_k(t_{\text{сб}}^j), \bar{X}_{k-1}(t_{\text{сб}}^{j-1}), \text{Alg}_{\text{AM}_{\text{UM}_k}}, \text{Alg}_{\text{AM}_{\text{UM}_{k-1}}}\right) \rightarrow \begin{cases} \min. \\ t_{\text{обп}}^j < h_{(k-1, k)} \\ h_{(k-1, k)} \gg 0 \end{cases}$$

где: $\text{Alg}_{\text{AM}_{\text{UM}_k}}$ – алгоритм функционирования k -го узла-агента;

$\text{Alg}_{\text{AM}_{\text{UM}_{k-1}}}$ – алгоритм функционирования $k-1$ узла-агента в дуге

$\{\text{UM}_{k-1}, \text{UM}_k\}$ орграфа $G(\text{OM}, \text{E})$;

Ограничения и допущения:

– значения $t_{\text{сб}}^j$, $t_{\text{пр}}^j$, $t_{\text{пер}}^j$ и $t_{\text{конф}}^j$ не оказывают существенного влияния на значение t_{delay}^j .

1.5 Выводы по главе

1. В разделе рассмотрены принципы организации и функционирования современных высоконагруженных распределенных вычислительных систем, применяемых в различных предметных областях. Выявлено, что основной особенностью подобных систем является масштабирование их структуры с одновременным усложнением функциональных возможностей, обусловленное ростом их пользовательской базы и увеличением сложности объектов, в рамках которых они функционируют. Рассмотрены структура и функциональная схема систем мониторинга подобных систем, для которой последние выступают в качестве объекта мониторинга. Определено, что структура современных систем мониторинга строится по схеме «агент-менеджер» и носит преимущественно централизованный или иерархический характер. На конкретных примерах

рассмотрена проблема функционирования систем мониторинга, связанная с наличием временного лага распространения нестационарной нагрузки по узлам мониторинга распределенных вычислительных систем, значение которого не совпадает с скважностью мониторинга этих узлов, что приводит к неэффективным управляющим воздействиям системы мониторинга на конкретные узлы мониторинга и снижает эффективность функционирования объекта мониторинга в целом.

Таким образом, на практике существует противоречие между необходимостью своевременного применения управляющих воздействий на узлах мониторинга, которые формирует система мониторинга и существующим математическим и программным обеспечением этой системы, время реакции которого может запаздывать относительно временного лага распространения нестационарной нагрузки на узлах мониторинга.

2. Рассмотрены существующие подходы к моделированию процесса мониторинга при нестационарной нагрузке. На основе анализа существующих подходов к моделированию такого процесса для различных предметных областей (мониторинга мультисервисных систем связи, вычислительные системы высокоскоростной обработки данных и др.) рассмотрены методы моделирования, базирующиеся на теории систем массового обслуживания, методах оценивания параметров нестационарных динамических систем, таких как фильтр Калмана и скрытые марковские модели. Определен базовый недостаток подобных подходов, связанный с возможностью моделирования преимущественно резких «всплесков» нагрузки, а также отсутствием механизмов ее прогнозирования на длительных периодах мониторинга.

Для условий моделирования процесса мониторинга крупномасштабных распределенных объектов мониторинга рассмотрены подходы, основанные на методах машинного обучения, базирующиеся на глубоких сетях доверия, автокодировщиках и рекуррентных нейронных сетях различного типа. Рассмотрен подход, связанный с использованием нейронных сетей долгой краткосрочной памяти для решения задач прогнозирования временных рядов.

3. В рамках рассмотренной структуры и функций системы мониторинга, а также особенностей нестационарной нагрузки на узлы объекта мониторинга, имеющей временной лаг распространения, определена цель проводимого исследования и дана формальная постановка научной задачи исследования, связанная с минимизацией времени задержки управляющего воздействия для заданного узла мониторинга (подмножества узлов) в условиях ограничений и допущений на параметры временного лага распространения нестационарной нагрузки на указанные узлы мониторинга.

ГЛАВА 2. РАЗРАБОТКА МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ МОНИТОРИНГА ДЛЯ ПРОГНОЗИРОВАНИЯ МОМЕНТОВ НЕСТАЦИОНАРНОЙ НАГРУЗКИ, ИМЕЮЩЕЙ ВРЕМЕННОЙ ЛАГ РАСПРОСТРАНЕНИЯ

В данной главе представлены подходы к разработке математического обеспечения распределенной системы мониторинга децентрализованного типа, обеспечивающей решение проблемы прогнозирования нестационарной нагрузки, имеющей временной лаг распространения по узлам объекта мониторинга.

Обоснованно выбрана архитектура такой системы мониторинга, основанная на децентрализованной одноранговой схеме, узлом которой является интеллектуальный агент мониторинга. Для указанного вида агента мониторинга разработана его модель, базирующаяся на структуре варианта рекуррентной нейронной сети с долгой краткосрочной памятью, для которой обоснованно выбрано количество слоев, определяющих глубину прогнозирования, а также структура элемента краткосрочной памяти, обеспечивающего пространственно-временную корреляцию вектора параметров нагрузки «соседних» узлов мониторинга и обеспечивающего накопление ретроспективных значений этих параметров с целью решения задачи прогнозирования их временного ряда.

2.1 Исследование подходов к организации децентрализованной архитектуры распределенной системы мониторинга на основе интеллектуальных агентов мониторинга

2.1.1 Анализ структурной и функциональной моделей систем мониторинга с централизованной структурой

В п. 1.2 была рассмотрена обобщенная архитектура существующих СМ, применяемых для современных крупномасштабных распределенных вычислительных систем различной функциональности. На основе проведенного

анализа было определено, что подобные СМ имеют преимущественно централизованную (единый центральный узел) или иерархическую (подмножество центральных узлов иерархической подчиненности) структуры. В качестве базовой схемы организации подобных СМ выступает схема «агент-менеджер» (рисунок 1.12). При этом для взаимодействия агентов, располагаемых на УМ с менеджером своего уровня иерархии (для иерархически организованных СМ) используются протоколы взаимодействия, специализированные для предметной области применения СМ. Например, в СМ инфокоммуникационных систем, а также МСС используются различные версии протокола сетевого управления SNMP [48, 49]. Для СМ систем распределенных вычислений характерно использование двух уровневой схемы «язык сериализации структур данных»-«протокол передачи данных». Например, реализация СМ Zabbix (рисунок 1.16) использует формат обмена данными JSON [50], применяемый поверх протоколов HTTP/HTTPS.

К достоинствам СМ с централизованной структурой стоит отнести:

- простоту программной или программно-аппаратной реализации узла-агента, связанную с его ограниченной функциональностью (рисунок 2.1);
- наличие единого центра управления структурой СМ, обеспечивающего общий каталог объектов (УМ, каналы связи, узлы-агенты). Так, например, архитектура СМ на основе протокола SNMP описывает структуру каталога MIB (Management Information Base) [51], используемую для хранения параметров всех объектов СМ.

Между тем, централизованная структура СМ, наряду с перечисленными достоинствами имеет ряд существенных недостатков, к которым можно отнести:

- сложность масштабирования СМ для крупномасштабных ОМ, состоящих из тысяч и более УМ. Эта сложность связана, либо с необходимостью существенного увеличения вычислительных возможностей узла-менеджера, способного в заданные временные рамки формировать требуемые управляющие воздействия, либо с необходимостью введения дополнительных уровней иерархии (менеджеров нижележащих уровней) (рисунок 2.2). В последнем случае для

множества частных МІВ, поддерживаемых указанными менеджерами, требуется введение процедур синхронизации и поддержки целостности данных мониторинга;

- увеличение трафика каналов управления, занимающего существенную часть общей полосы пропускания каналов ОМ.

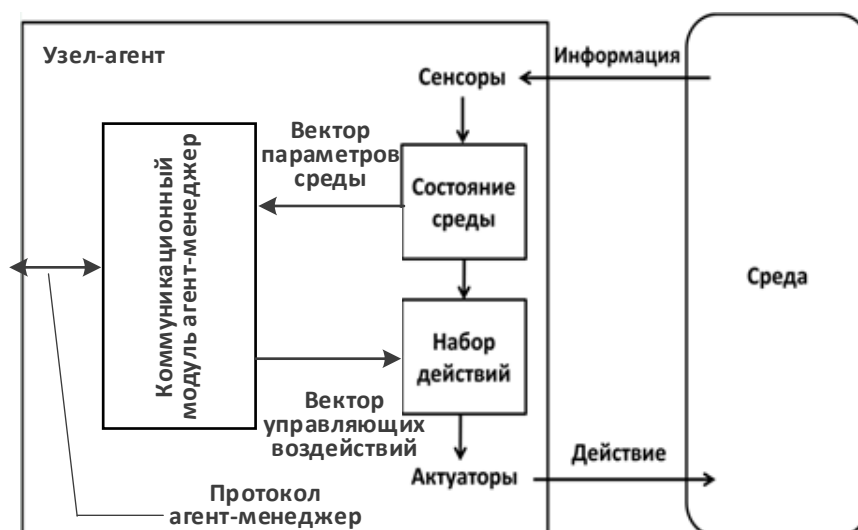


Рисунок 2.1 – Структурная схема типового узла-агента СМ с централизованной структурой

Из рисунка 2.1 видно, что узел-агент СМ с централизованной структурой рассматривает подконтрольный УМ (или подмножество подконтрольных УМ) как управляемую среду, которая формирует поток информации о своем состоянии и изменяет это состояние под действиями, поступающими от СМ. Соответственно, поток информации о состоянии УМ узел-агент воспринимает набором сенсоров, а управляющие воздействия передает УМ набором актуаторов (исполнительных механизмов). Для хранения и преобразования в требуемый формат поступающих от УМ параметров, а также набора действий, поддерживаемых актуаторами, узел-агент может содержать локальный вариант МІВ. Взаимодействие с узлом-менеджером осуществляется с использованием коммуникационного модуля, который формирует протокольные блоки данных, содержащие вектор параметров

УМ, и интерпретирует протокольные блоки данных, содержащие вектор управляющих воздействий, поступающий от узла-менеджера.

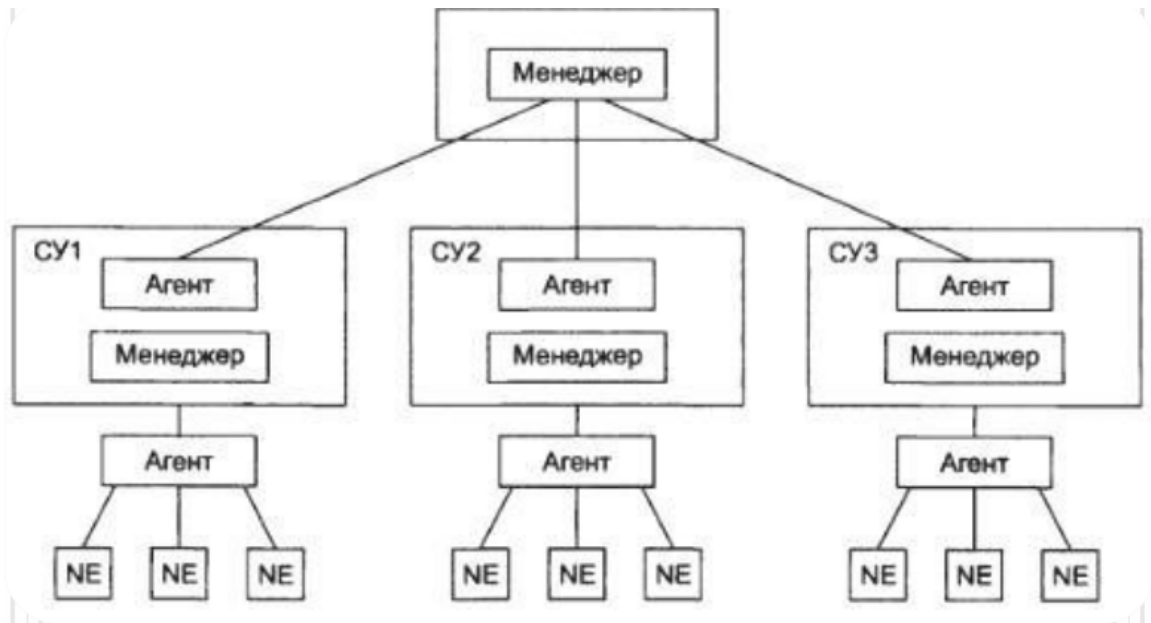


Рисунок 2.2 – Пример СМ с двухуровневой иерархической структурой (узлы-менеджеры второго уровня являются узлами-агентами для центрального узла-менеджера)

Поскольку логика формирования управляющих воздействий для множества узлов-агентов в СМ сосредоточена на узле-менеджере, то при масштабировании СМ (увеличении количество узлов-агентов), а также в случае увеличения интенсивности потока векторов параметров от имеющихся узлов-агентов требуется пропорциональное увеличение вычислительной мощности модуля, формирующего управляющие воздействия. Обычной практикой в данном случае является программная реализация такого модуля в виде многопоточного приложения в операционной системе узла-менеджера, где отдельный поток с фиксированной логикой связан с одним подконтрольным узлом-агентом на основе его уникального идентификатора, например, UUID. Подобный вариант масштабирования, основанный на распараллеливании типовой функциональной логики, является наименее ресурсозатратным и, в настоящее время поддерживается подавляющим большинством фреймворков распределенных вычислений. При этом

существенным недостатком подобной реализации СМ является отсутствие учета корреляционных связей различного типа, возникающих в подмножестве УМ. Как было указано в п. 1.4 в настоящем исследовании выдвигается гипотеза о наличии пространственно-временной корреляции между векторами параметров нагрузки УМ в условиях временного лага ее распространения. Распространение параметров нестационарной нагрузки осуществляется в рамках некоторого взвешенного ациклического графа, узлами которого являются подмножество УМ, веса ребер отражают значение временного лага распространения нагрузки, а направления ребер указывают вектор распространения этой нагрузки.

Для учета подобной пространственно-временной корреляции в СМ централизованного типа, функциональная логика узла-менеджера должна поддерживаться соответствующими алгоритмами межпоточной синхронизации данных, ориентированными на учет пространственно-временных связей, и соответствующими им структурами данных – буферами межпоточного взаимодействия.

Реализация подобной функциональной логики при этом будет узкоспециализированной для каждого ОМ, в рамках которых разворачивается СМ. Кроме того, централизованный учет пространственно-временной корреляции задастую требует априорных знаний о законе распределения моментов нестационарности, что, в большинстве случаев, является практически невыполнимым.

2.1.2 Анализ структурной и функциональной моделей систем мониторинга с децентрализованной структурой на основе мультиагентных систем

В качестве альтернативного варианта организации СМ, в рамках которой возможен учет пространственно-временной корреляции между векторами параметров нагрузки на УМ предлагается рассмотреть схему с децентрализованным управлением, реализуемую в рамках МАС - многоагентных систем (MAS, Multi-agent system) [52, 53, 54].

В [54] под МАС понимается распределенная система программных и/или программно-аппаратных агентов, совместно функционирующих с целью достижению некоторого результата. Под агентом в МАС при этом понимается автономная структура, взаимодействующая с подмножеством себе подобных с использованием различных механизмов.

За исключением простейшей схемы поведения агентов на основе механизма «стимул-реакция», при которой для отдельно взятого агента остальное подмножество агентов выступает в качестве среды, наряду с УМ, базовой схемой поведения агентов в МАС является совещательная схема [52].

Реализация указанной схемы в рамках традиционной структуры узла-агента (рисунок 2.1) невозможна, в силу отсутствия у подобных агентов необходимой функциональности. МАС с совещательной схемой взаимодействия формируются на основе интеллектуальных агентов (ИА).

Обобщенная структурная схема ИА представлена на рисунке 2.2.

В [53] под ИА понимается узел-агент, функции которого обеспечивают следующие базовые свойства:

1. Реактивность - возможность анализа среды, с которой осуществляется взаимодействие и реакция на изменения ее свойств путем выполнения действия, поддерживающих достижение целевой функции.

2. Проактивность - по-другому, инициативное поведение, связанное с анализом предполагаемого изменения среды и выполнения действий упреждающей реакции на такие изменения.

3. Социальность – возможность взаимодействия с другими сущностями среды (в первую очередь другими ИА) для достижения целевой функции.

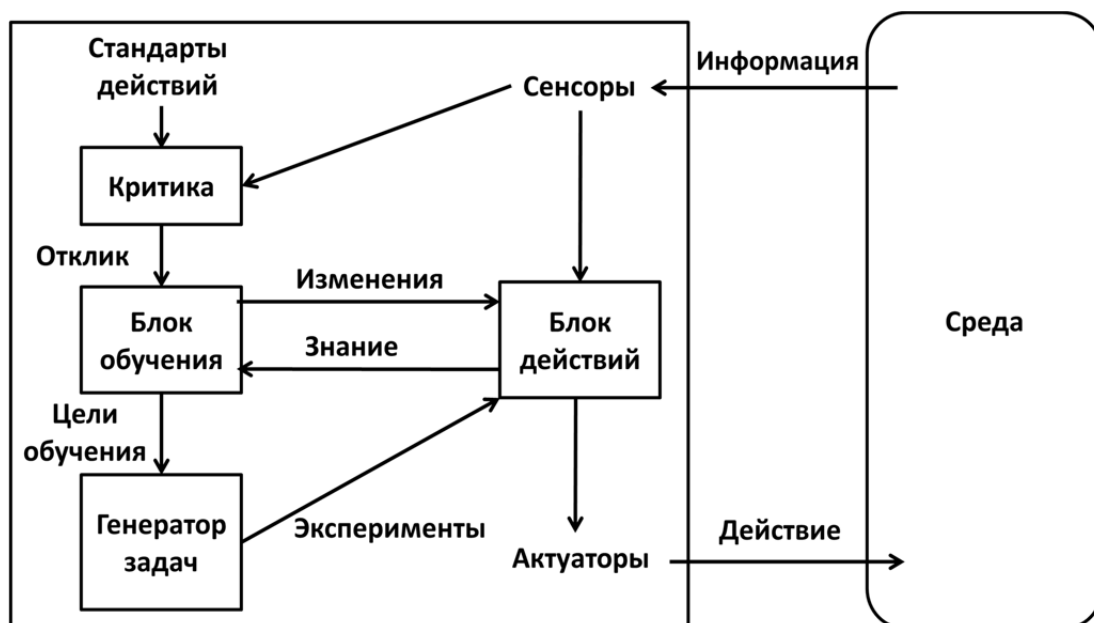


Рисунок 2.3 – Обобщенная структурная схема интеллектуального агента (ИА)

Из рисунка 2.3 видно, что основой ИА является блок обучения, обеспечивающий динамическое формирование набора действий в зависимости полученной от среды информации и экспериментального (тестового) выполнения вариантов задач, решение которых обеспечивает достижение целевой функции. При этом в рамках МАС процесс обучения ИА распространяется и на свойство социальности, при рассмотрении подмножества ИА, как часть среды, с которой взаимодействует обучающийся ИА.

То есть, в широком смысле взаимодействие ИА в МАС означает динамическое формирование, в простейшем случае, двусторонних и в более общем случае многосторонних отношений между внутри подмножества ИА.

В узком смысле взаимодействие ИА выступает в роли механизма обучения ИА, обеспечивающее возможность реализации свойства проактивности.

В [56] при рассмотрении проблемы взаимодействия ИА выделяются следующие классы задач взаимодействия:

- кооперация (сотрудничество);
- конкуренция (конфликт);
- компромисс (учет целевой функции других ИА);

- конформизм (отказ от своей целевой функции в пользу целевой функции других ИА);

- автономность (отказ от взаимодействия).

Решение конкретной задачи, относящейся к тому или иному классу, в общем случае, требует от разработчика МАС выполнения следующих действий:

1. Определение класса задачи взаимодействия.
2. Определение подмножеств взаимодействующих ИА.
3. Определение ролевой модели взаимодействия и распределение функциональных ролей между ИА.
4. Синтез формальной модели взаимодействия ИА в рамках полученных ограничений.
5. Определение стратегий поведения ИА.
6. Выбор или синтез протокола взаимодействия ИА.

При этом протокол взаимодействия ИА является подобием протокола взаимодействия менеджера и агента в централизованной схеме СМ, то есть основан на обмене сообщениями с использованием специализированных языков взаимодействия (ACL, Agent Communication Language). К известным реализациям языков ACL можно отнести языки KQML (Knowledge Query and Manipulation Language) и FIPA-ACL (Foundation for Intelligent Physical Agents), обобщенное представление которых дается в [52, 54]. Для реализации межагентного взаимодействия в случаях поддержки ИА разных стандартов ACL в рамках их структур данных формируется база данных онтологии (БДО) поддерживаемого ИА объекта (в рамках исследования – УМ), которая является инвариантной к формату ACL. Сообщения на ACL разных стандартов трансформируются на уровне БДО. В качестве стандарта представления данных в БДО обычно используется одна из версий языка OWL (Ontology Web Language) [55]

Обобщенное представление указанного процесса представлено на рисунке 2.4.

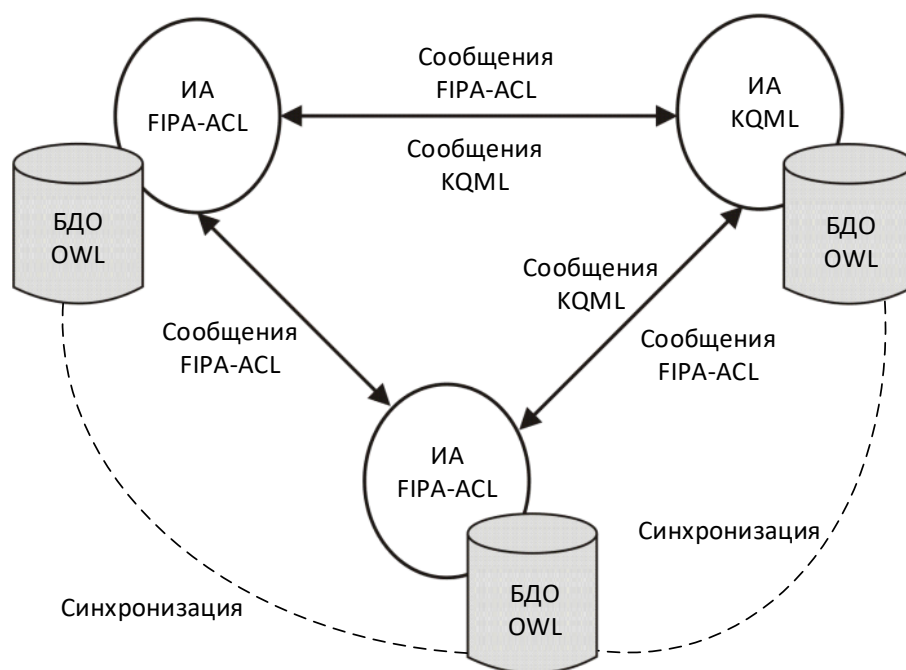


Рисунок 2.4 – Обобщенное представление процесса межагентного взаимодействия на основе разных стандартов ACL

Кроме механизма межагентного взаимодействия для представленных выше классов задач взаимодействия, исключая автономное поведение (ИА, важной задачей является решение проблемы согласования данных и действий подмножества ИА, обеспечивающей достижение общей надёжности МАС в условиях неточной или ошибочной информации, поступающей она сенсоры ИА от среды (рисунок 2.3). В теории распределенных вычислений подобная проблема именуется проблемой достижения консенсуса [57]. В ряде исследований, посвященных МАС, указанная проблема именуется взаимным информационным согласованием (ВИС) подмножества агентов [58, 59].

Таким образом, при разработке структуры и функций СМ с децентрализованным управлением на основе МАС необходимо разработать формальную модель ИА, далее – ИА мониторинга (ИАМ), учитывающую, как рассмотренные выше особенности обобщенной схемы ИА, так и задачи взаимодействия, и ВИС подмножества ИА.

2.2 Разработка нейросетевой модели интеллектуального агента мониторинга, на основе рекуррентной нейронной сети с длинной цепью элементов краткосрочной памяти

Выбор децентрализованной схемы СМ, основанной на множестве ИАМ требует модификации обобщенной схемы СМ с централизованным управлением (рисунок 1.11) путем преобразования ее в схему однорангового взаимодействия подмножества ИАМ. Результат такой модификации представлен на рисунке 2.5.

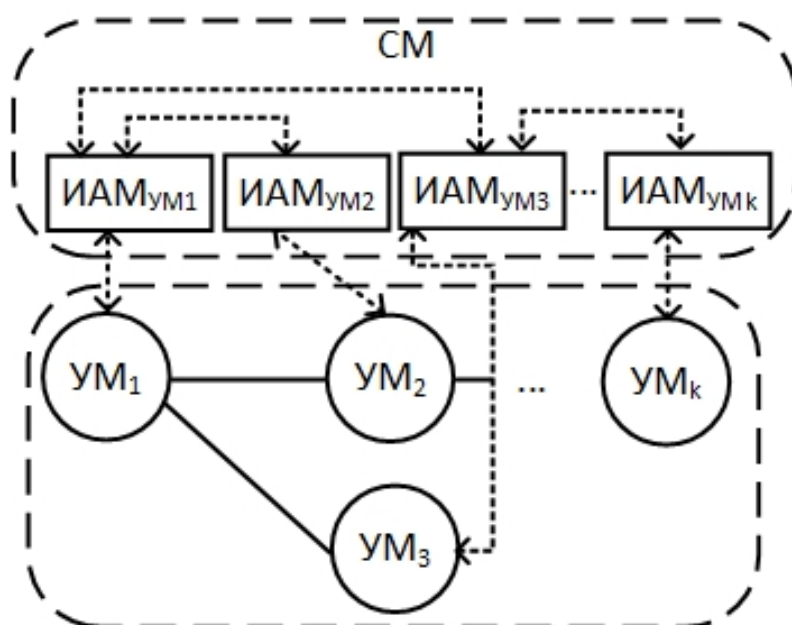


Рисунок 2.5 – Обобщенная структура системы мониторинга децентрализованного типа на основе подмножества интеллектуальных агентов мониторинга

Из рисунка 2.5 следует, что множество ИАМ размещается на множестве УМ, образуя взаимодействующую пару $УМ_k \leftrightarrow ИАМ_{УМk}$. $УМ_k$ при этом является частью среды, с которой взаимодействует $ИАМ_{УМk}$, получая вектор параметров нагрузки $\bar{X}_k(t_{сб}^j)$ k -го УМ в момент времени $t_{сб}^j$. Другой частью среды, с которой $ИАМ_{УМk}$ осуществляет взаимодействие, является подмножество $ИАМ_{УМ}$,

формируемое в соответствии с взвешенным орграфом распространения вектора параметров нагрузки $G(OM, E), \langle \{UM_i, UM_k\}, \prec \rangle \in E, UM \in OM$.

Таким образом, в отличие от традиционной для MAC Grid-схемы взаимодействия агентов (взаимодействие по типу «каждый с каждым»), в исследовании предлагается сократить количество связей взаимодействия ИАМ. То есть, в качестве источника анализа ИАМ_{УМ_к} использует только значения вектора параметров $\bar{X}_k(t_{c\bar{c}}^j)$ и значения векторов параметров, формируемых УМ_{к-1} и УМ_{к+1}, которые связаны с УМ_к временным лагом h их распространения согласно виду орграфа $G(OM, E)$. Можно считать, что подмножество $\{UM_{k-1}, UM_{k+1}\}$ является источником ретроспективных значений векторов параметров $\bar{X}_{k-1}(t_{c\bar{c}}^j)$ и $\bar{X}_{k+1}(t_{c\bar{c}}^j)$, и, в свою очередь, УМ_к входит в подмножество источников ретроспективных параметров для других соседних УМ. В теории графов такой порождённый подграф орграфа $G(OM, E)$, состоящий из всех вершин, сопряжённых УМ_к и всех рёбер, соединяющих две такие вершины, называется *окрестностью* (Neighbourhood) вершины УМ_к и обозначается $N_{G(OM, E)}(UM_k)$ [60].

2.2.1 Выбор варианта нейросетевой модели интеллектуального агента мониторинга

Предложенная на рисунке 2.5 схема организации СМ с децентрализованной структурой на основе множества $\{ИАМ_{УМ1}, \dots, ИАМ_{УМк}\}$ предполагает, что структурная схема ИАМ_{УМ} соответствует рисунку 2.3, из которого следует, что наиболее важным компонентом ИАМ является блок обучения, обеспечивающий интеллектуальный анализ значений векторов параметров нагрузки параметров

$\bar{X}_{k-1}(t_{cб}^j)$, $\bar{X}_k(t_{cб}^j)$, $\bar{X}_{k+1}(t_{cб}^j)$. Поскольку требуется выполнять ретроспективный анализ временного ряда (вектор $\bar{X}_k(t_{cб}^j)$ является вариантом временного ряда), следовательно согласно общей классификации нейросетевых моделей (рисунок 2.6) [61] следует рассматривать класс рекуррентных нейронных сетей (Recurrent neural network, RNN) [61, 62].

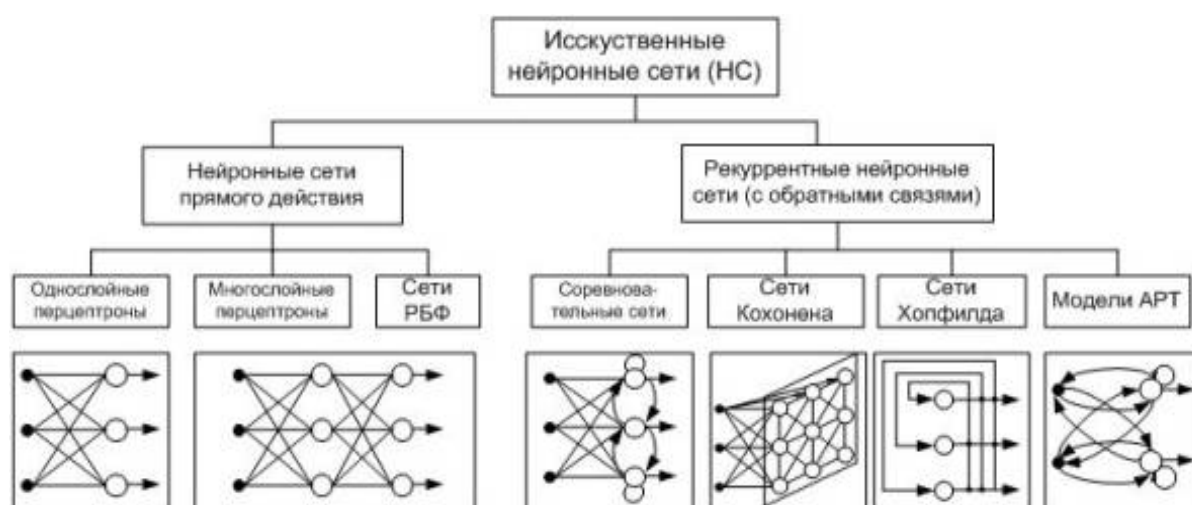


Рисунок 2.6 – Обобщенная классификация нейронных сетей

Выбор RNN обусловлен тем, в структуре этих сетей выходы нейронов последующих слоев соединены со входами нейронов предшествующих слоев. Подобная структура RNN обеспечивает учет результатов преобразования выходного вектора параметров на предыдущем этапе для обработки входного вектора параметров на следующем этапе функционирования сети (рисунок 2.7). Из рисунка следует, что фактически, RNN следует рассматривать, как несколько копий слоя (layer) L_t , активация которого происходит в момент времени t , который передает параметры последующей копии – слою L_{t+1} . Такой подход называется учетом темпоральных (временных) зависимостей. Благодаря этому RNN нашли широкое применение в решении задач прогнозирования (машинный перевод) и генеративных моделях (создание музыки, изображений).

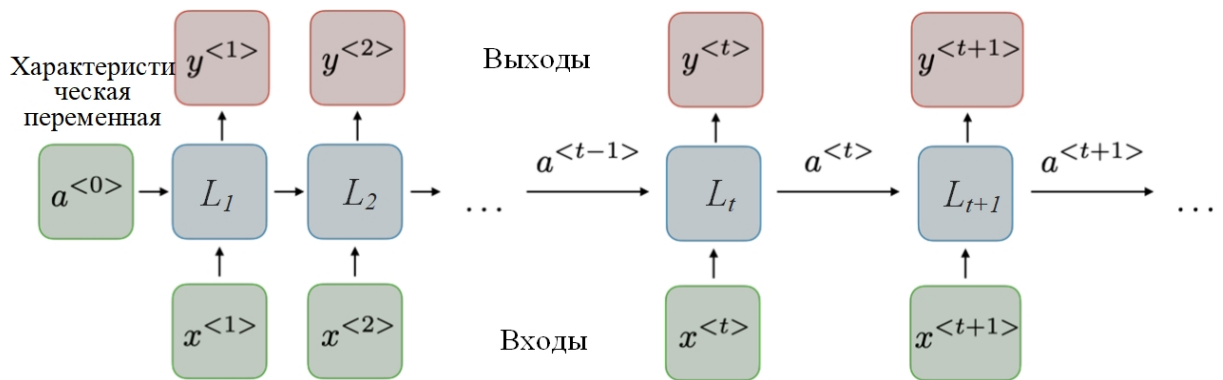


Рисунок 2.7 – Обобщенная структура рекуррентной нейронной сети

Для каждого момента времени t значения выхода $y^{<t>}$ слоя L_t и его характеристической переменной $a^{<t>}$ (в ряде источников именуется «скрытым» состоянием) определяются выражениями 4 и 5.

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (4)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y), \quad (5)$$

где g_1 и g_2 - характеристические функции (функции активации – обычно логистический сигмоид σ (далее *sigmoid*) или гиперболический тангенс далее *tanh*), а W_{aa} , W_{ax} – весовые коэффициенты соответствующих параметров активации и входа, а b_a – вектор смещения этих параметров.

С учетом выражений 4 и 5, структура слоя L_t может быть представлена следующей схемой (рисунок 2.8).

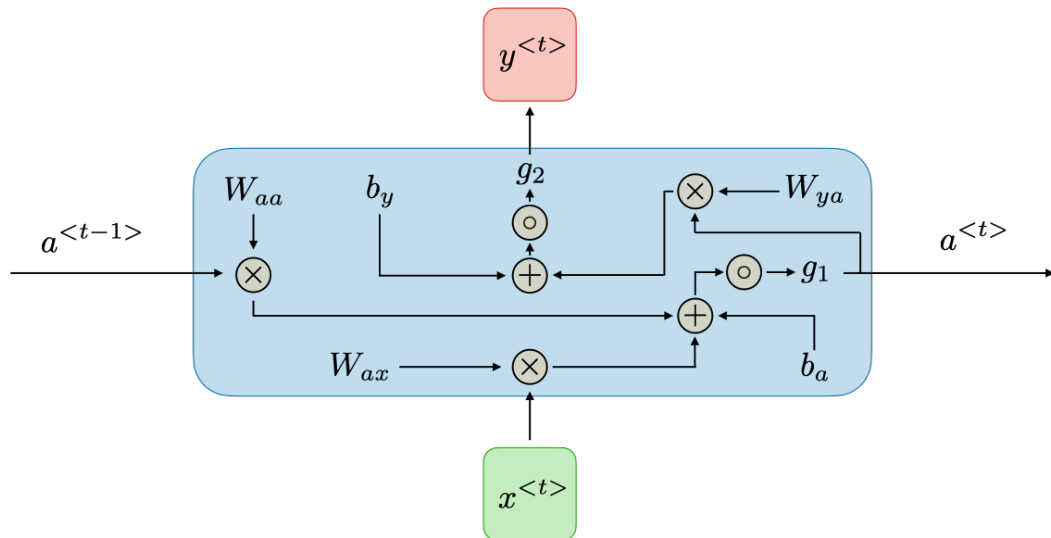


Рисунок 2.8 – Обобщенная структура слоя L_t рекуррентной нейронной сети

Существенным недостатком рассмотренной структуры RNN, затрудняющей ее использования в рамках поставленной цели исследования, является учет темпоральных зависимостей только близлежащих (соседних) слоев (так называемая краткосрочная память). При этом, для решения проблемы прогнозирования момента нестационарности нагрузки в подмножестве УМ в момент времени $t+1$ требуется учет зависимостей в более ранние моменты времени их распространения по вершинам орграфа $G(OM, E)$. Другими словами, для более точного прогноза для UM_k необходимо получение данных из окрестности $N_{G(OM, E)}(UM_k)$. Основной проблемой невозможности использования RNN классической архитектуры в задачах долгосрочного прогнозирования является проблема «исчезающих» (vanishing) или «взрывающихся» (exploding) градиентов в ходе выполнения задачи обратного распространения ошибки (ОРО) методами градиентного спуска, наиболее применимыми для RNN-сетей. Поскольку функции активации в задаче ОРО являются линейными и при этом представляют отношение производных от значения ошибки за промежуток T , то даже небольшие изменения значений весовых коэффициентов W_{aa} и W_{ax} будут приводить к экспоненциальному росту или убыванию значений градиента, что объясняется наличием точек бифуркации (рисунок 2.9) [63]. Соответственно, это приводит к

сложности получения зависимостей на длительных временных промежутках – получения достоверного контекста из прошлых состояний сети.

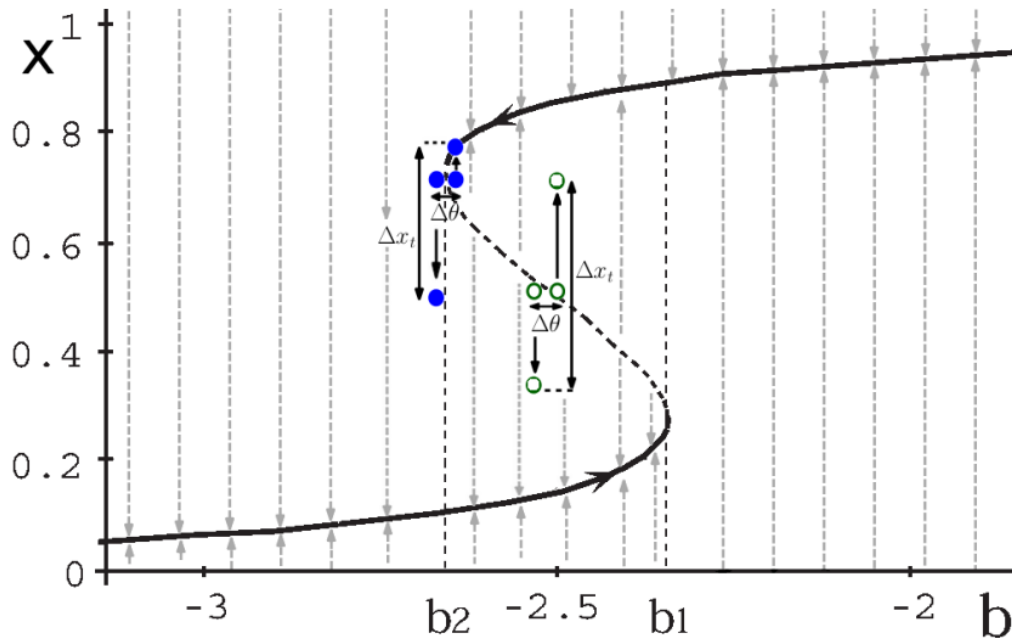


Рисунок 2.9 – Пример наличия точек бифуркации в градиенте слоя RNN с фиксированным весовым коэффициентом $W_{ax} = 0,5$ в заданном диапазоне вектора смещения параметра b_a (источник [63])

Частично, проблему получения контекста в долгосрочных промежутках T решают специальные архитектуры классических RNN, например, такие как RNN Джордана и Элмана [65]. Особенностью этих архитектур RNN является введением обратных связей, или от выходов сети (RNN Джордана) или от выходов нейронов скрытого слоя (RNN Элмана) (рисунок 2.10). Такая структура позволяет учитывать предшествующие состояния процесса, что в свою очередь ведет к накоплению информации о его контексте и обеспечивает выработку более точного решения. Основной проблемой таких архитектур является сложность процесса обучения, связанная с необходимостью возвращения значений градиента к самому началу последовательности.

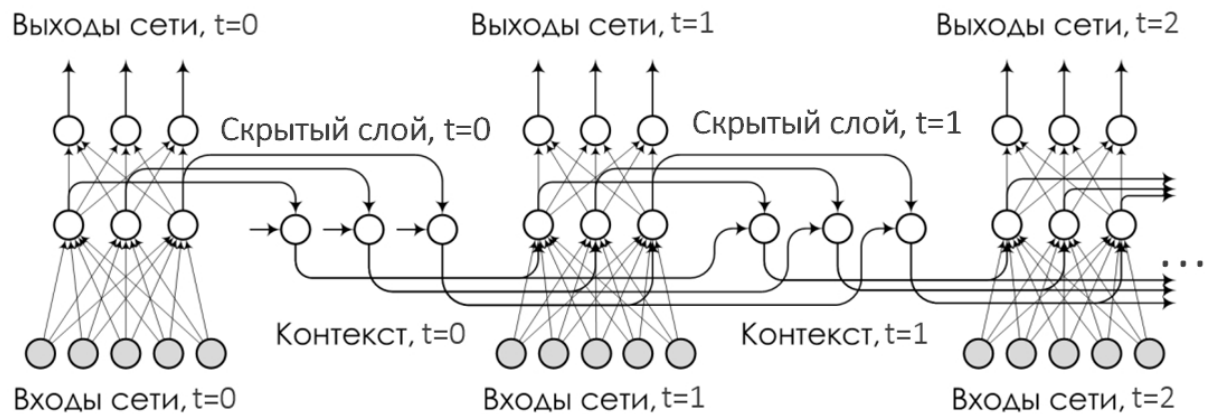


Рисунок 2.10 – Обобщенная структура RNN Элмана

Поскольку, согласно формальной постановке задачи исследования (п. 1.4) требуется минимизировать время задержки управляющего воздействия t_{delay}^j , а время обучения сети является одной из составляющих $t_{\text{обр}}^j$ в его составе, то для устранения этого недостатка в исследовании предлагается использование сети LSTM – длинной цепи элементов краткосрочной памяти (Long short-term memory), которая является разновидностью рассмотренных выше архитектуры RNN [62, 64]. Основным отличием LSTM от RNN с обратными связями является возможность их использования при прогнозировании временных рядов, в частности, в случаях, когда значения требуемых параметров разделены временными лагами с неопределённой продолжительностью и границами. Это достигается тем, что архитектура LSTM наряду (или, в ряде случаев, вместо) с рассмотренными на рисунках 2.8 и 2.9 элементами слоя L_t содержит элемент (ячейку) памяти $\text{Cell}_{\text{LSTM}}$ – рекуррентный модуль со специальной структурой (рисунок 2.11) [67], имеющий возможность запоминать (хранить) значения параметров темпоральных зависимостей $x^{<t>}$ и $a^{<t>}$. Комбинируя слои из $\text{Cell}_{\text{LSTM}}$ -ячеек, сохраняющих значения темпоральных зависимостей, как на коротких ($t \pm 1$), так и на длинных ($t \pm n$) промежутках времени можно увеличивать, как глубину прогноза за счет накопления ретроспективных данных, так и его точность за счет ретроспективного анализа влияния этих параметров на выходное значение. Наличие значений,

хранимых ячейкой $\text{Cell}_{\text{LSTM}}$, позволяет не применять для них функцию активации, обеспечивая целостность хранимых значений во времени. А это, в свою очередь, означает, что в процессе решения задачи обратного распространения ошибки, градиент (штраф) не исчезает (эффект размывания). Из рисунка 2.11 видно, что структура ячейки $\text{Cell}_{\text{LSTM}}$ дополнительно к характеристической переменной $a^{<t>}$ содержит специальную переменную $c^{<t>}$, именуемую состоянием $\text{Cell}_{\text{LSTM}}$. Эта переменная отображает темпоральное состояние $\text{Cell}_{\text{LSTM}}$ – информацию о сохраненных, накопленных или удаленных в момент времени $t-1$ в ячейке данных. Управлением переменной $c^{<t>}$ выполняется специальными функциями-фильтрами, именуемые в архитектуре LSTM «шлюзами» (gate). Классический подход к разработке $\text{Cell}_{\text{LSTM}}$ [63] определяет «шлюзы» четырех типов: входной (input gate), выходной (output gate), забывания (forget gate), «шлюз» нового «кандидата» в состояние $\text{Cell}_{\text{LSTM}}$ (candidate cell state gate). Из рисунка 2.9 видно, что «шлюзы» входной, выходной и «шлюз» забывания определены на функции sigmoid (выражения 6, 7, 8), а «шлюз кандидата» определен на функции tanh (выражение 9).

$$i_t = \sigma(W_{xi}x_t + W_{ai}a_{t-1} + b_i) \quad (6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ao}a_{t-1} + b_o) \quad (7)$$

$$f_t = \sigma(W_{xf}x_t + W_{af}a_{t-1} + b_f) \quad (8)$$

$$c'_t = \tanh(W_{xc}x_t + W_{ac}a_{t-1} + b_{c'}) \quad (9)$$

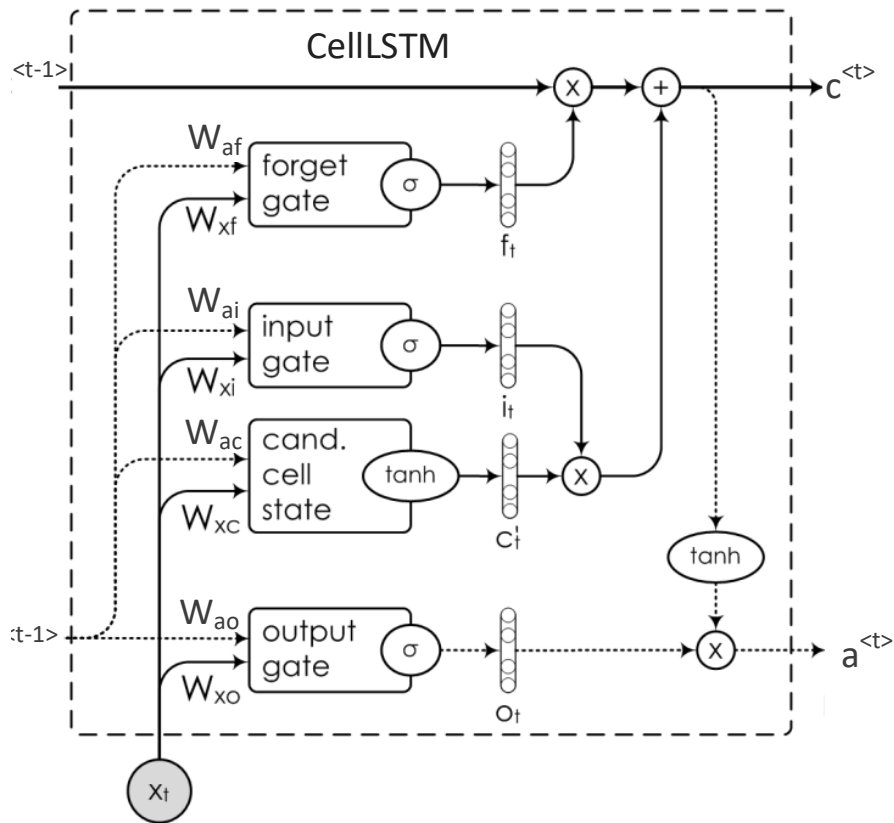


Рисунок 2.11 – Обобщенная структура ячейки памяти $Cell_{LSTM}$ сети LSTM

Также из рисунка 2.11 следует, что текущее состояние $c^{<t>}$ зависит от того, будет ли предыдущее состояние $c^{<t-1>}$ замещено новым состоянием «кандидатом» (выражение 10).

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad (10)$$

Кроме того, значение характеристической переменной $a^{<t>}$ также зависит от текущего состояния $c^{<t>}$ (выражение 11).

$$a_t = o_t \odot \tanh(c_t) \quad (11)$$

Рассмотренная выше структура $Cell_{LSTM}$ является базовой и в процессе совершенствования архитектуры LSTM модифицировалась для применимости в конкретных условиях. Например, в условиях, когда значение выходного «шлюза» равно нулю («шлюз» закрыт) поведение сети LSTM будет не зависеть от состояния $Cell_{LSTM}$ (вырождение в RNN-структуру). Для устранения этого в функцию

базовых шлюзов вводится коэффициент состояния ячейки, реализуемый матричным перемножением.

2.2.2 Структура нейросетевой модели интеллектуального агента мониторинга для решения задачи прогнозирования моментов возникновения нестационарной нагрузки

Прогноз значений временного ряда в момент времени t с использованием LSTM выполняется для момента времени $t+1$, используя предыдущие значения временного ряда $(t-1, t-2, \dots, t-\tau)$, где значение τ определяет глубину ретроспективного «погружения» - количество слоев LSTM. На входы слоя LSTM поступают значения вектора $x(t) = (x_{t-\tau+1}, x_{t-\tau+2}, \dots, x_t)$, а выходным при этом является прогнозное значение x_{t+1} . Обучающая выборка для такой LSTM сети формируется не только на основе значений временного ряда параметров $УМ_k$, подконтрольного ИАМ $_k$, но и параметров $УМ_{k-1}$ и $УМ_{k+1}$, связанных с $УМ_k$ временным лагом h их распространения. То есть, одно подмножество $\{УМ_{k-1}, УМ_{k+1}\}$ является источником ретроспективных параметров, а в свою очередь, $УМ_k$ входит в подмножество источников ретроспективных параметров для других соседних УМ (рисунок 2.5). Таким образом, значения анализируемых параметров от множества всех УМ находятся в пространственно-временной корреляции.

Учет подобной корреляции предлагается реализовать в элементе $Cell_{LSTM}$. Для определения корреляции между значениями параметров в разных $УМ_k$ $Cell_{LSTM}$ может быть представлен корреляционной матрицей A_t . Так, в общем случае, для ОУ, состоящего из множества узлов $УМ_{OU} = \{УМ_1, УМ_2, \dots, УМ_m\}$, матрица A_t определяется, как $A_t: УМ_{OU} \times УМ_{OU}$. Состояния элементов множества $УМ_{OU}$ при этом находятся, как во временной корреляции (текущие значения

параметров UM_m и значения параметров в предыдущий момент мониторинга), так и в пространственной корреляции (УМ, являющиеся соседними - подмножество $\{UM_{m-1}, UM_m, UM_{m+1}\}$) текущему узлу по вектору распространения пиковой нагрузки. То есть, значения элементов матрицы являются A векторами, характеристики которых связаны, как со временем момента мониторинга t , так и интервалом S_M , определяющим скважность цикла мониторинга.

$$A_{t,S_M} = Cr(A_{t+S_M}, A_{t+2S_M}, \dots, A_{t+NS_M}), \quad (12)$$

где Cr - кросскорреляционная функция, $A_{t+iS_M} = [x_1, x_2, \dots, x_m]^t$ - вектор параметров x_j УМ ($1 \leq j \leq m$) за i -й временной интервал S_M .

Коэффициент участия $l_{j,k}$ параметров j -го УМ в k -й точке мониторинга за временной интервал S_M , как $|j-k| \times S_M$. Тогда интервальную кросскорреляционную функцию можно определить, как:

$$l_{j,m}(S_M) = Cr(X(t), X'(t+S_M)), \quad (13)$$

где $X(t)$ - временной ряд j -го УМ, а $X'(t+S_M)$ - временной ряд m -го УМ. Таким образом, в момент времени t на вход элемента $Cell_{LSTM}$ подается вектор параметров, имеющий тесную связь с состоянием УМ в момент времени $t-1$, и динамика изменения значений элементов матрицы A_t будет зависеть, как от текущего значения момента мониторинга t , так и от временного интервала S_M . Сохраняемые в $Cell_{LSTM}$ значения применяются в качестве входных параметров слоев L_t сети LSTM, как на этапе ее обучения, так и при решении задачи прогнозирования параметров UM_j в момент времени $t+1$.

В обобщенном виде структура предлагаемой нейросетевой модели для прогнозирования $i+q$ параметров мониторинга UM_j и глубины прогноза $t+r$ представлена на рисунке 2.12 [64].

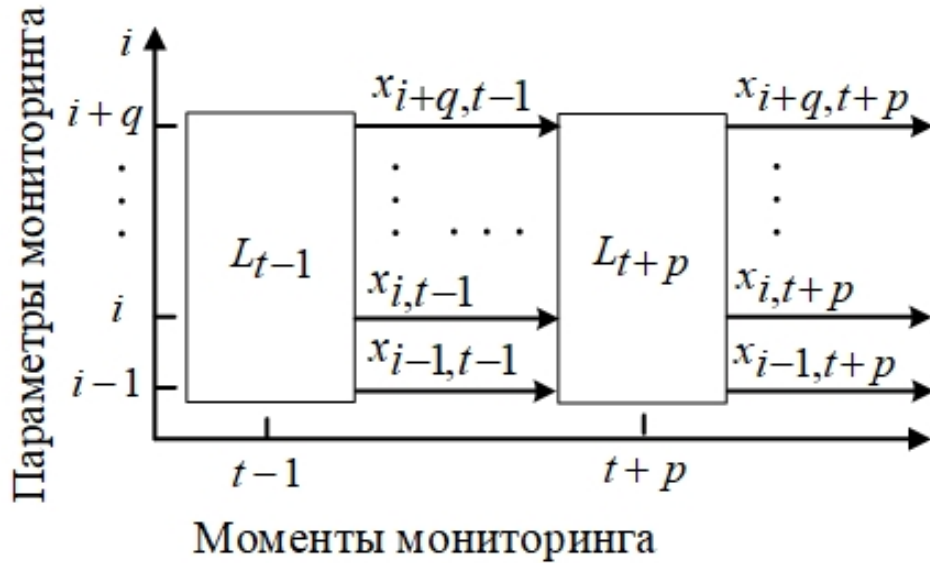


Рисунок 2.12 – Обобщенный вид предлагаемой нейросетевой модели ИАМ

Количество слоев L_t определяет глубину прогнозирования $T_{Пр}$, а значения Δt_i определяют скважность моментов мониторинга S_M :

$$T_{Пр} = \sum_{i=1}^p \Delta t_i \quad (14)$$

Из рисунка 2.12 видно, что горизонтальное измерение указывает на изменения во временной области (моменты мониторинга), а вертикальное – на вектор параметров точек наблюдения ($\bar{X}_k(t_{сб}^j)$ – параметры мониторинга). То есть предлагаемая структура сети LSTM представляет собой пространственно-временную сеть. Вертикальная ось указывает на индексы точек наблюдения, которые находятся в восходящем порядке. После назначения индексов определяются пространственные расстояния по пространственной оси. Горизонтальная ось указывает на точки наблюдения во временном пространстве.

При этом значения Δt_i должны корректироваться путем минимизации суммы квадратичных ошибок выходных значений слоя. Исходя из анализа [64, 65], для оценивания целевого показателя (п. 1.4) предлагается использовать три критерия: средняя абсолютная ошибка (MAE), средняя квадратичная ошибка (MSE) и средняя

относительная ошибка (MRE) (выражение 15). Каждая из которых определяется следующим образом:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\tilde{\varphi}_i - \varphi_i|,$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (\tilde{\varphi}_i - \varphi_i)^2, \quad (15)$$

$$MRE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\tilde{\varphi}_i - \varphi_i}{\varphi_i} \right|,$$

где $\tilde{\varphi}_i$ – данные прогноза, а φ_i – данные измерений.

Согласно выражениям (15), MAE и MSE более чувствительны к необработанным данным, следовательно, MRE больше подходит в качестве оценочного критерия.

В развернутом виде элементы структуры слоя L_t в момент мониторинга от $t-2$ представлены на рисунке 2.13 [66].

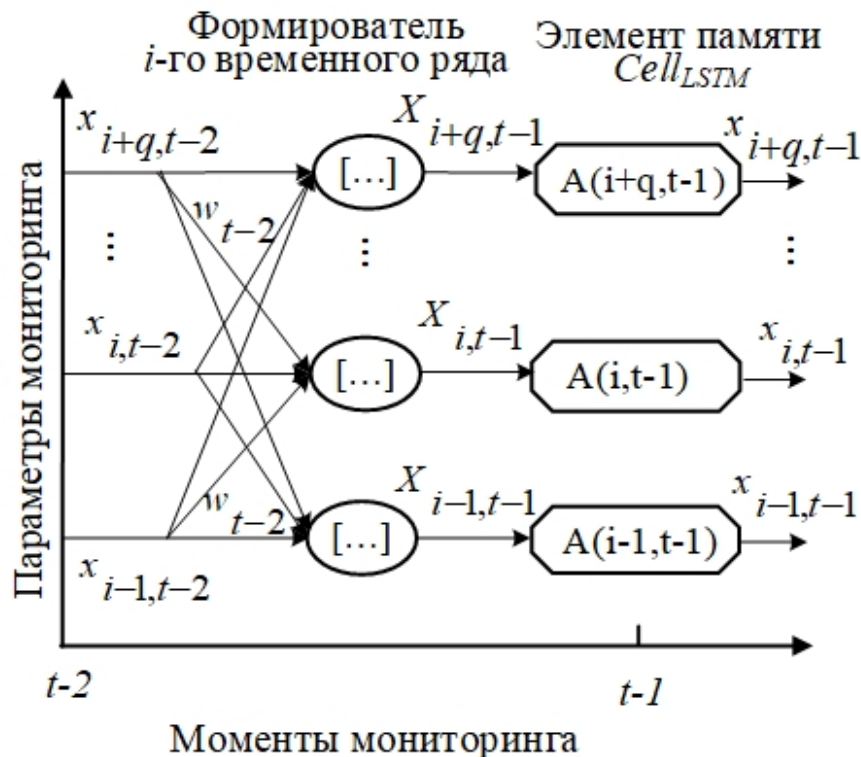


Рисунок 2.13 – Структура слоя предложенной L_{t-1} LSTM-сети

2.3 Моделирование процесса обучения разработанной нейросетевой модели интеллектуального агента мониторинга

Поскольку данные об необходимом количестве слоев для разработанной LSTM-сети априори неизвестны, однако результат ее функционирования оказывает воздействие на подконтрольный УМ, в качестве базового метода обучения (настройки параметров ее слоев) было предложено использование класса методов обучения с подкреплением (Reinforcement Learning). Так как отличительной особенностью ИАМ, согласно теории мультиагентных систем, является его автономность, то для решения задачи такой настройки предлагается использовать алгоритм конфигурирования структуры LSTM-сети, который реализуется в составе программного обеспечения ИАМ.

Определим процесс обучения с подкреплением LSTM-сети, как *эпизод* (episode), описываемым конечным марковским процессом принятия решений (МППР), представленным кортежем $\langle S, A_s, P_s, \pi \rangle$ (состояния, действия, вероятности перехода из состояния в состояние), где π - стратегия, некоторая функция сопоставления действий состояниям [70], поскольку он является повторяющимся взаимодействием ИАМ со «средой» - подконтрольным УМ.

Так как настройка конфигурации LSTM-сети производится для двух гиперпараметров: количества слоев и количества эпох обучения, то определим, что текущая конфигурация содержит множество $n \geq 1$ слоев, и для получения этой текущей конфигурации требуется k эпох обучения. Тогда любое из состояний МППР $s \in S$, исключая начальное, связано с этими гиперпараметрами. Текущую конфигурацию LSTM-сети определим как множество $\{l, j \cdot e\}$, где $l \subset n$ - число слоев (layers), e - количество эпох (epoch) обучения, необходимых для этого числа слоев.

$j \cdot e$ – произведение элементов числового ряда $j = \left\{ 1, 2, \dots, \frac{k}{e} \right\}$ на целое число e , определяющее значение нужного для обучения числа эпох. Тогда множество

состояний МППР S для нее определим, как $s \in S_{l,j \cdot e}$. Для $s \in S_{l,j \cdot e}$ определим множество действий:

$$A_{s_{l,j \cdot e}} = \{a_{s_{l,j \cdot e} \& s_{l,(j+1) \cdot e}}, a_{s_{l,j \cdot e} \& s_{l+1,e}}\} \quad (16)$$

Действие $a_{s_{l,j \cdot e} \& s_{l,(j+1) \cdot e}}$ - сравнение конфигураций $\{l, j \cdot e\}$ и $\{l, (j+1) \cdot e\}$.

Если точность прогнозирования в конфигурации $\{l, (j+1) \cdot e\}$ будет выше, чем в предыдущей, то выполняется переход в состояние $s' = s_{l,(j+1) \cdot e}$ в противном случае состояние $s = s_{l,j \cdot e}$ остается неизменным.

Действие $a_{s_{l,j \cdot e} \& s_{l+1,e}}$ - определение факта повышения точности прогнозирования при добавлении нового слоя (изменение конфигурации $\{l, j \cdot e\}$ на $\{s+1, e\}$). Переход в состояние $s' = s_{l+1,e}$ происходит при повышении точности прогнозирования с новым слоем. В противном случае состояние $s = s_{l,j \cdot e}$ остается неизменным.

Определим вероятность перехода из состояния s в состояние s' при выполнении действия a_s как $P(s'|s, a_s)$, а время, затрачиваемое на выполнение этого действия, как $T(s'|s, a_s)$. Тогда при выполнении действия $a_{s_{l,j \cdot e} \& s_{l,(j+1) \cdot e}}$ с переходом в состояние $s' = s_{l,(j+1) \cdot e}$ (то есть, итерации обучения сети с l слоями и количеством эпох обучения $(j+1) \cdot e$) требуется время:

$$T_1(s'|s, a_{s_{l,j \cdot e} \& s_{l,(j+1) \cdot e}}) = t_1 \cdot (j+1) \cdot e, \quad (17)$$

где t_1 - время выполнения одной эпохи обучения для сети с l слоями.

При выполнении действия $a_{s_{l,j \cdot e} \& s_{l+1,e}}$ сеть с конфигурацией $\{l+1, e\}$ требует дообучения на e эпохах, для чего требуется время:

$$T_2(s' | s, a_{s_{1,j}e} \& s_{l+1,e}) = e \cdot t_1 + 1 \quad (18)$$

Диаграмма переходов рассматриваемого МППР представлена на рисунке 2.14.

Исходя из выражений 16-18, оптимальной стратегией π будет та, в которой выбранная конфигурации LSTM-сети максимизирует точность прогнозирования (выход LSTM-сети) при минимизации значений времен T_1 или T_2 . В качестве метода решения этой задачи предлагается использование методов динамического программирования, в частности, модификации алгоритма итерации по значениям (Value Iteration), базой которого является уравнение Беллмана [69].

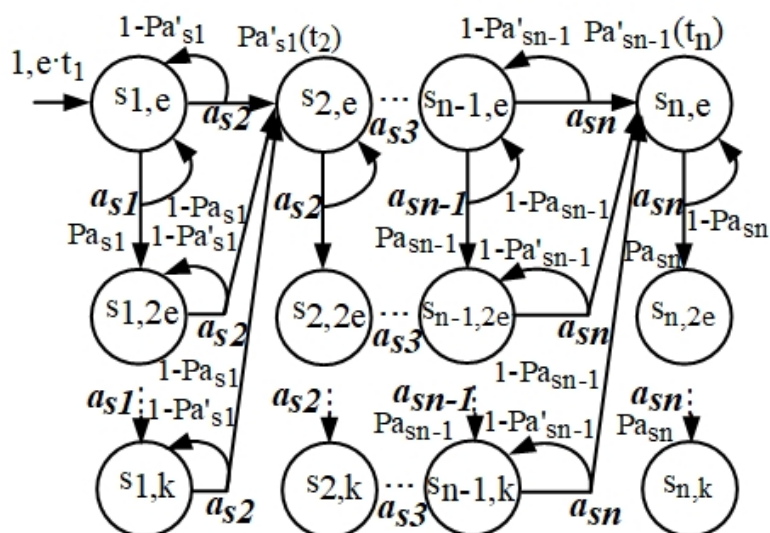


Рисунок 2.14 – Диаграмма переходов марковского процесса принятия решений, описывающего эпизод обучения LSTM-сети

В соответствии с этим для введем функцию качества действия (Q -функция)

$$q_i^\pi(s, a_s):$$

$$q_i^\pi(s, a_s) = \sum_{s'} P(s' | s, a_s) \cdot (T(s' | s, a_s) + \gamma \cdot v_{i-1}(s')), \quad (18)$$

где $v_{i-1}(s')$ - функция ценности состояния s' для стратегии π на $i-1$ итерации (v -функция), а $0 \leq \gamma \leq 1$ - коэффициент обесценивания (discount rate). Поскольку

для рассматриваемого МППР все затраты ИАМ, связанные с обучением сети накапливаются, то принимается условие $\gamma=1$. Исходя из принципа оптимальности Беллмана, функция ценности состояния s в текущей итерации i определяется как минимум функции $q_i^\pi(s, a_s)$:

$$v_i(s) = \min q_i^\pi(s, a_s) \quad (19)$$

Использование предлагаемого подхода к моделированию процесса обучения нейросетевой модели ИАМ позволяет перейти к синтезу итерационного алгоритма динамической настройки параметров LSTM-сети, который будет формировать ее оптимальную конфигурацию с точки зрения точности прогнозирования моментов возникновения нестационарной нагрузки на подконтрольном УМ.

2.4 Выводы по главе

1. В разделе выполнен процесс моделирования структуры нейросетевой модели, являющейся основой блока обучения интеллектуального агента мониторинга. На основании рассмотренного в главе 1 процесса мониторинга, а также развернутого анализа структуры и особенностей моделей на базе рекуррентных нейронных сетей и их модификаций делается выбор структуры варианта рекуррентной нейронной сети, обеспечивающей возможность решения задачи прогнозирования значений временного ряда параметров нагрузки на узлы мониторинга, расположенные в окрестности подконтрольного узла мониторинга и связанные временным лагом распространения моментов нестационарности этой нагрузки.

2. Рассмотрены типовые структуры рекуррентных нейронных сетей, обеспечивающие учет долговременных временных зависимостей. Определены их особенности и недостатки, ограничивающие применимость для определенной в главе 1 цели исследования.

Для указанных особенностей, определяющих цель исследования, в качестве основы нейросетевой модели обосновано выбран класс рекуррентных сетей с

долгой краткосрочной памятью (LSTM-сетей). Предложена матричная структура ячейки памяти LSTM-сети, обеспечивающая учет пространственно-временной корреляции параметров нагрузки, связанных с временным лагом ее распространения, и являющейся матрицей связности гиперпараметров LSTM-сети и накопленных значений параметров нагрузки узлов мониторинга в окрестности подконтрольного узла мониторинга с учетом характеристик временного ряда распространения моментов нестационарности нагрузки.

3. Для предложенной структуры LSTM-сети рассмотрены подходы к процессу ее обучения (настройке конфигурации сети), основанные на представлении этого процесса в виде марковского процесса принятия решений, что позволило решить задачу определения времен обучения сети в рамках одной эпохи, а также ее дообучения с использованием варианта метода динамического программирования – алгоритма итерации по значениям, что в дальнейшем позволяет перейти к синтезу итеративного алгоритма динамической настройки параметров разработанной LSTM-сети.

ГЛАВА 3. РАЗРАБОТКА ОБОБЩЕННОГО АЛГОРИТМА ДИНАМИЧЕСКОЙ НАСТРОЙКИ ПАРАМЕТРОВ РАЗРАБОТАННОЙ НЕЙРОСЕТЕВОЙ МОДЕЛИ В ЗАДАННОЙ ТОЧКЕ МОНИТОРИНГА И ОБОБЩЕННОГО АЛГОРИТМА ВЗАИМНОГО ИНФОРМАЦИОННОГО СОГЛАСОВАНИЯ ИАМ

В данной главе выполнен анализ существующих алгоритмов и методов настройки гиперпараметрических конфигураций рекуррентной нейронной сети типа LSTM. На основе этого анализа разработан обобщенный алгоритм динамической настройки параметров нейросетевой модели на основе LSTM сети, функционирующей в составе множества ИАМ_{УМК}.

Разработан обобщенный алгоритм взаимного информационного согласования подмножества ИАМ_{УМК} посредством многоадресной рассылки сообщений, основанной на парадигме «Издатель-Подписчик». Представлены структурные схемы алгоритмов процедур передачи и приема сообщений при многоадресной рассылке сообщений. Для реализации управления потоком сообщений обоснованно выбран протокол децентрализованного управления Pastry, на основе которого разработан алгоритм наложенного управления сообщениями.

3.1 Разработка обобщенного алгоритма динамической настройки параметров разработанной нейросетевой модели в заданной точке мониторинга

Для оптимизации нейросетевой модели (п. 2.2.2) при прогнозировании временного ряда необходимо выбрать оптимальную гиперпараметрическую конфигурацию, включающую количество скрытых слоев и количество эпох обучения, которые влияют не только на общее время обучения, но и на точность прогнозирования. В общем случае в процессе обучения с учителем изменение указанной конфигурации выполняется вручную для каждого этапа обучения. Таким образом этот процесс является итерационным до тех пор, пока модель не обеспечит заданную точность прогнозирования.

Указанный подход к обучению мало применим к многоузловым децентрализованным структурам, так как требует наличие узла, который используется «учителем» нейросетевой модели. Кроме того, для крупномасштабных многоузловых ОМ процесс переобучения является трудоемким, особенно с точки зрения временных затрат. При этом существует ряд исследований, связанных с автоматизацией процесса конфигурирования гиперпараметров нейросетевых моделей.

В их основе лежит разработка алгоритмов, которые заданному критерию принимают решение о необходимости реконфигурации гиперпараметров модели. Например, в [75] для автоматической настройки рекуррентных сетей используют BOA (Bayesian optimization approach), а в [76] – NMM (Neural minimization methods).

Метод BOA был разработан для нахождения оптимального значения целевой функции черного ящика. В нашем случае, целевая функция черного ящика относится к модели LSTM, а оптимальное значение относится к настройке гиперпараметров. Неопределенность целевой функции для еще не оцененных значений моделируется как априорное распределение вероятностей, которое отражает наши представления о поведении функции. После сбора оценок функций (т.е. точности прогнозирования модели LSTM при определенной настройке гиперпараметра) априорное значение обновляется для формирования апостериорного распределения по целевой функции. Затем апостериорное распределение используется для построения функции сбора данных, которая выберет наиболее многообещающее значение (т.е. наиболее многообещающую настройку гиперпараметра) для следующей оценки. Описанный выше процесс повторяется в направлении достижения оптимума.

В свою очередь, NMM является методом оптимизации нелинейных функций. Он не требует какой-либо производной информации, что делает его подходящим для задач с негладкими функциями. NMM минимизирует целевую функцию, генерируя исходный симплекс на основе предопределенной вершины, а затем вычисляет значение функции в каждой вершине. В нашем случае, вершина - это набор значений, присвоенных гиперпараметрам LSTM, предопределенная вершина

- это набор гиперпараметров LSTM по умолчанию, а вычисление функции заключается в получении ошибки предсказания (prediction error) для LSTM, обученного с определенным набором данных при определенной гиперпараметрической конфигурации. Затем последовательно выполняется последовательность преобразований над симплексом, направленная на уменьшение значений функций в его вершинах. Возможные преобразования включают в себя отражение, расширение, сжатие и усыхание. Вышеуказанный процесс завершается, когда выборочное стандартное отклонение значений функции текущего симплекса падает ниже некоторого допустимого значения.

Методы BOA и NMM применимы только в стандартных рекуррентных сетях, где входными данными являются собственные значения нейронной сети, на которых модель автономно обучается. Для разработанной нейросетевой модели ИАМ_{УМК} они являются мало применимыми, так как входными данными для обеспечения в процессе обучения этой модели пространственно-временной корреляции будут использоваться не только значения параметров нагрузки заданного ИАМ_{УМК}, но и значения, полученные от соседних ИАМ на заданных участках временного ряда.

Таким образом, в рамках исследования предлагается разработать обобщенный итерационный алгоритм динамической настройки параметров (ДНП), специализированный для предлагаемой в главе 2 нейросетевой модели. При этом, поскольку каждый эпизод обучения с подкреплением предлагается представить марковским процессом принятия решений получения оптимальной стратегии $\pi(s)$ действий ИАМ_{УМК} при его взаимодействии со «средой» (п. 2.3), то для решения этой оптимизационной задачи предлагается использование алгоритма итерации по значениям (Value iteration) – варианта метода динамического программирования, а в качестве критерия принятия решения использование функции ценности состояния (выражение 19).

Для обеспечения повышения точности прогнозирования значений параметров состояния ОМ в условиях нестационарной нагрузки, предлагается,

чтобы отслеживание характеристик каждого УМ производилось СМ в режиме реального времени, а каждая комбинация ИАМ-период СМ, то есть каждый ИАМ в различный временной период времени, имел свою собственную модель LSTM [71].

Структурная схема разработанного итерационного обобщенного алгоритма представлена на рисунке 3.1.

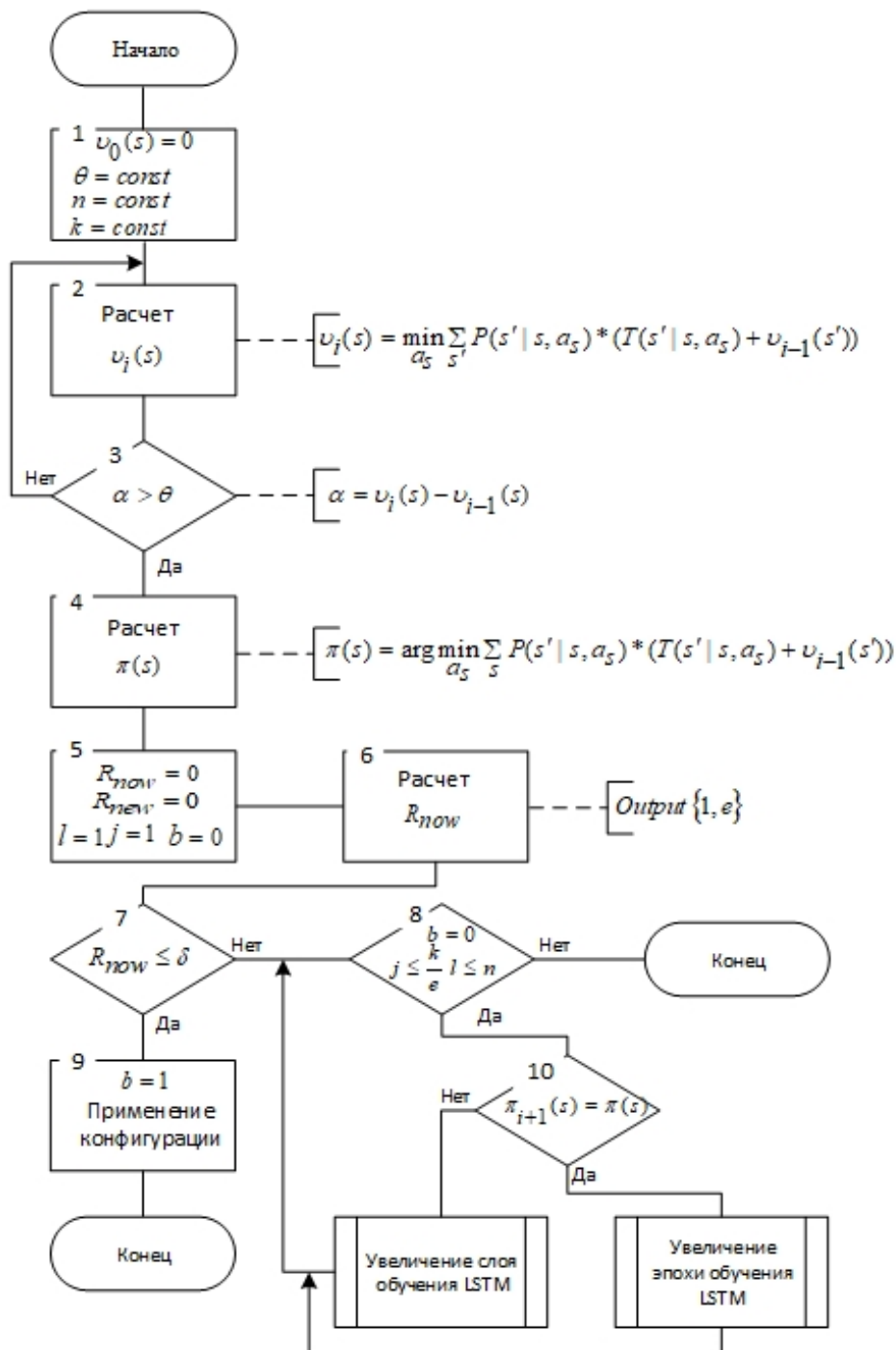


Рисунок 3.1 – Структурная схема алгоритма ДДНП LSTM-сети ИАМ

Из рисунка видно, что в блоке 3 выполняется проверка положительности значения разности текущими и ожидаемыми временными затратами (переменная α). Для этого вводится пороговое значение θ . При выполнении этого условия текущая конфигурация LSTM-сети считается оптимальной и для текущего состояния s в блоке 4 вычисляется значение оптимальной (приводящей к минимуму временных затрат) стратегии $\pi(s)$. Однако оптимальная с точки зрения времени стратегия $\pi(s)$ может не приводить к заданной точности прогнозирования. Для решения этой задачи алгоритм продолжает поиск конфигурации $\{l, j \cdot e\}$ удовлетворяющей, как стратегии $\pi(s)$, так заданной пороговой точности прогнозирования (константа δ в блоке 5). Для этого вводятся переменные R_{now} и R_{new} (блок 5), определяющие выходное значение сети для текущей конфигурации и абсолютную вероятностную ошибку выходного значения для новой конфигурации сети соответственно, а также бинарная переменная b проверки условия в блоке 8. Далее алгоритм итерационно изменяет текущую конфигурацию $\{l, j \cdot e\}$ пока одновременно не будут выполнены условия блоков 3 и 10. При их выполнении переменная принимает значение $b=1$ и полученная конфигурация сети применяется для решения задачи прогнозирования. Этапы увеличения количества слоев l и эпох обучения e являются отдельными алгоритмами и представлены предопределенными процедурами.

Соблюдение всех предложенных действий обеспечивает минимизацию временных затрат, но это не гарантирует, что полученная конфигурация сможет достичь удовлетворительной точности прогноза. С другой стороны, продолжение поиска конфигурации и ее использование для переобучения LSTM может привести к повышению точности прогнозирования, но это может занять очень много времени. Для решения этой проблемы в алгоритме предусмотрено ограничение по поиску конфигураций до уровня обеспечения удовлетворительной или максимально возможной точности прогнозирования. Более подробно этот процесс описан в главе 2 в п. 2.3.

Более того, для каждого отдельного ИАМ в СМ модели LSTM могут полностью отличаться друг от друга. Если учитывать то, что часы пиковых нагрузок на УМ в сети проявляются в рабочее время, то каждый ИАМ будет иметь 2 собственные LSTM модели (рабочее и не рабочее время). Тогда, общее количество моделей LSTM будет $2n$, где n - это общее количество ИАМ в СМ в крупномасштабной сети с нестационарной нагрузкой. Для эффективной настройки LSTM для каждой из комбинаций ИАМ-период $2n$, параллельно, мы расширяем ДНП в распределенном и параллельном виде, получая децентрализованный ДНП (ДДНП). Он использует набор вычислительных узлов для разделения рабочей нагрузки настроек. Пока доступен вычислительный узел, ДДНП запрашивает у него настройку модели LSTM для ИАМ-периода. Таким образом, настройка LSTM для всех $2n$ ИАМ-периодов может выполняться параллельно [72].

3.2 Разработка обобщенного алгоритма взаимного информационного согласования интеллектуальных агентов мониторинга

Во 2 главе в п. 2.2 на рисунке 2.5 представлена и подробно описана предлагаемая обобщенная структура СМ децентрализованного типа на основе подмножества ИАМ_{УМК}. Очевидно, что сокращение количества связей взаимодействия ИАМ положительно влияет на уменьшение времени обмена данными между множеством ИАМ_{УМК}, что обеспечивает повышение эффективности процесса дообучения LSTM-сети.

Предлагается подход, основанный на взаимном информационном согласовании множества ИАМ_{УМК}, использующем наложенное управление многоадресной рассылкой сообщений, содержащих данные о моментах возникновения нестационарной нагрузки. Таким образом, процедура взаимного информационного согласования множества ИАМ_{УМК} может быть представлена двумя взаимосвязанными процедурами, расположенными на разных логических уровнях их выполнения (рисунок 3.2) [73].

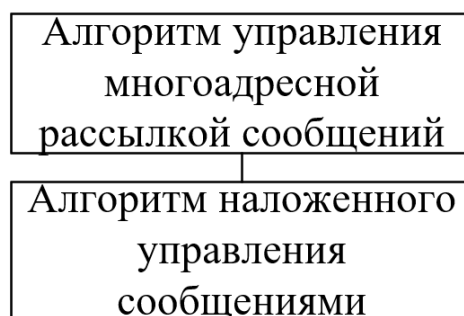


Рисунок 3.2 – Состав predetermined процедур обобщенного алгоритма взаимного информационного согласования ИАМ_{УМК}

3.2.1 Разработка алгоритма управления многоадресной рассылкой сообщений

Каждый из множества ИАМ_{УМК} в СМ заинтересован в получении ретроспективных данных о моментах возникновения нестационарной нагрузки на соседних с ним ИАМ. Обобщение таких данных в рамках обучающей выборки LSTM-сети ИАМ_{УМК} позволит распознать моменты возникновения нестационарной нагрузки в рамках пространственно-временной корреляции между подмножествами территориально или логически соседних ИАМ_{УМК}, что в свою очередь обеспечит возможность решения задачи прогнозирования временного ряда, соответствующего временному лагу распространения нестационарной нагрузки.

С целью повышения оперативности обмена такими данными между подмножествами ИАМ_{УМК} предлагается использовать децентрализованную систему уведомлений о событиях, основанную на парадигме «Издатель-Подписчик» (Publish-Subscribe), широко применяемую в крупномасштабных распределенных системах, которая обеспечивает эффективную многоадресную рассылку на уровне приложений и способна масштабироваться для большого числа ИАМ_{УМК}. Одним из конкретных примеров реализации такой парадигмы представлен в [70].

Согласно этой парадигме любой ИАМ_{УМК} может создать тему (Topic), являясь ее издателем (Publisher), а затем другие ИАМ_{УМК} могут зарегистрировать

свой интерес теме и стать ее подписчиками (Subscribers). ИАМ_{УМк} могут публиковать сообщения о событиях, создавать и подписываться на множество тем, а темы могут иметь множество издателей и подписчиков. При этом публиковать сообщения о событиях в теме могут только ИАМ_{УМк} с соответствующими разрешениями, например подмножество ИАМ-соседей, находящихся в окрестности подконтрольного ИАМ_{УМк}.

Исходя из этого каждый ИАМ_{УМк} может реализовывать функции (рисунок 3.3):

- издателя темы (topic publisher);
- подписчика на тему (topic subscriber);
- корневого узла многоадресной рассылки (message.root);
- узла дерева многоадресной рассылки (message.node).

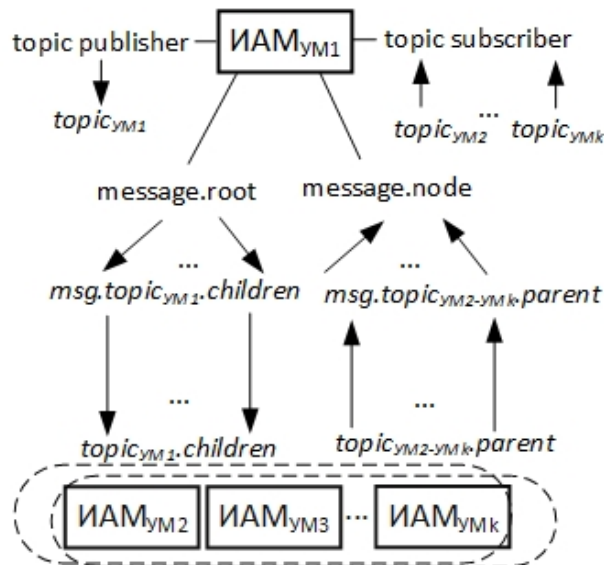


Рисунок 3.3 – Функции ИАМ_{УМк}, которые узел реализует в ходе многоадресной рассылки сообщений

ИАМ_{УМ1}, формирующий тему topic_{УМ1}, является ее издателем и одновременно может быть корневым узлом дерева многоадресной рассылки (message.root). Листьями (message.node) этого дерева является подмножество ИАМ_{УМ2-УМк} - topic_{УМ1}.children, выполнившие функцию SUBSCRIBE=topic_{УМ1}. Данные о дереве рассылки темы хранятся в виде циклического двусвязного списка

«parent-children» на узле `message.root`. Количество листьев регулируется функциями `SUBSCRIBE` и `UNSUBSCRIBE`. Дерево многоадресной рассылки существует то тех пока на тему `topicУМ1` подписан хотя бы один ИАМ_{УМк}. Очевидно, что каждый ИАМ_{УМк} может формировать (`PUBLISH`) более чем одну тему `topicУМк.children` и одновременно может быть подписан (`SUBSCRIBE`) на более чем одну тему `topicУМ1-УМк-1.parent` [74].

В рамках предложенной схемы многоадресной рассылки сообщений были разработаны алгоритмы создания темы, подписки и отписки от темы, рассылки сообщений темы и прием сообщений темы узлом-подписчиком (рисунок 3.4).

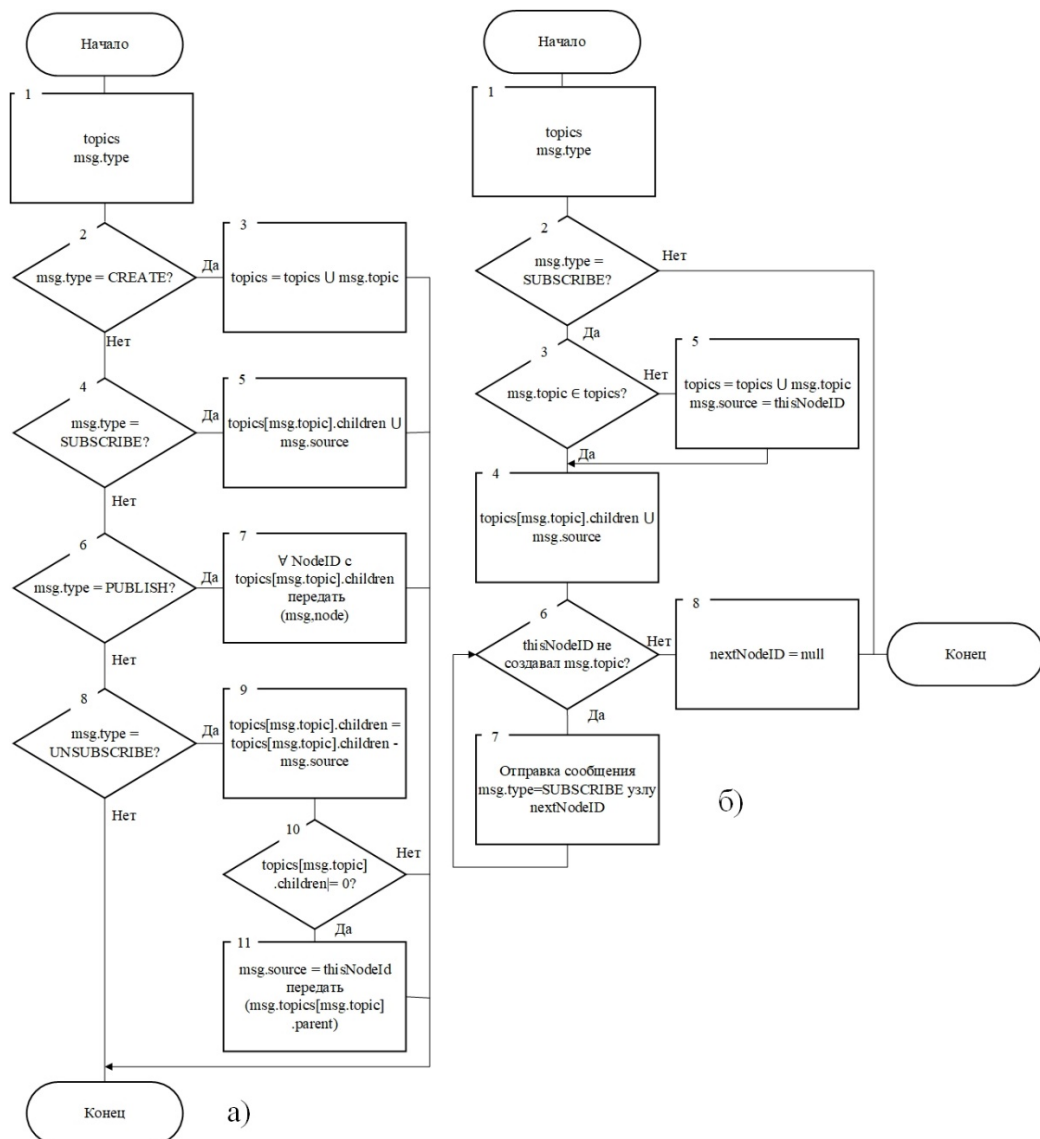


Рисунок 3.4 – Структурные схемы алгоритмов процедур ИАМ_{УМк} по управлению передачей а) и приемом б) сообщений при многоадресной рассылке

3.2.2 Разработка алгоритма наложенного управления сообщениями

Из представленных на рисунке 3.4 структурных схем алгоритмов видно, что адресация при обмене сообщениями родительских и дочерних ИАМ_{УМК} в пределах дерева многоадресной рассылки осуществляется на основе некоторого идентификатора узла NodeID. При этом связность списка узлов «предок-потомок» реализуется использованием двух типов NodeID: thisNodeID – идентификатор узла, инициализирующего отправку сообщения любого типа (msg.type), nextNodeID – идентификатор «соседнего» узла, выбранная мера близости к которому имеет минимальное значение. Очевидно, что если ИАМ_{УМК} формирует подписку на тему и одновременно является подписчиком на одну и более тем, то значение его NodeID будет использовано и для thisNodeID и для nextNodeID. Указанное решение позволяет реализовать многоадресную рассылку сообщений с точки зрения пространственно-временной корреляции подмножеств ИАМ_{УМК}.

При этом важной задачей для децентрализованной системы является выбор метода и реализация алгоритма наложенного управления сообщениями, минимизирующего число переходов между узлами и не допускающего эффекта лавинной рассылки сообщений (flood messaging). С этой целью в качестве базового метода управления предлагается рассматривать метод, основанный на критерии близости узлов (proximity routing), в частности – «выбор соседа поблизости» (proximity neighbor selection). Этот метод основан на концепции формирования для узлов управления сообщениями двух типов пространств: физического (территориального расположения узлов) и логического (пространства идентификаторов узлов).

Алгоритмы, реализующие указанный метод, основаны на формировании пространства идентификаторов узлов и выявления узлом, отправляющим сообщение, одного или более узлов-соседей, используемых для следующего шага, которые логически ближе всего соответствуют ключу сообщения (msg.topic) в пространстве идентификаторов узлов. То есть, среди всех возможных последующих шагов в пространстве идентификаторов узлов алгоритм выбирает,

либо ближайший узел в физическом пространстве узлов, либо узел, обеспечивающий компромисс между метриками близости в физическом и логическом пространствах.

В основе разрабатываемого алгоритма наложенного управления сообщениями лежит протокол децентрализованного управления, архитектура которого основана на децентрализации с распределенной структурой (полная децентрализация). Основными представителями протоколов структурированной оверлейной сети, использующие DHT, являются Chord, Pastry и Kademlia. Особенности этого подхода и характеристика протоколов подробно описаны в [58].

3.2.2.1 Обоснование выбора протокола децентрализованного взаимодействия интеллектуальных агентов мониторинга

Для выбора протокола децентрализованного взаимодействия был произведен анализ производительности управления сообщениями структурированного оверлея в DHT, а именно Chord, Pastry и Kademlia. В данном исследовании анализ проводился в соответствии с метриками производительности, такими как: эффективность управления сообщениями, количество шагов поиска и задержка успешного поиска.

Моделирование структурированных оверлейных протоколов производилось с использованием OMNeT++ с INET и модулем OverSim. OverSim - это система моделирования P2P с открытым исходным кодом, которая имеет трехуровневую архитектуру, разработанную с использованием языка NED. OverSim был описан как мощный инструмент для оценки и разработки приложений P2P.

На прикладном уровне реализуется управление сообщениями на основе ключей (KBR), которая периодически отправляет тестовые сообщения на случайные узлы и записывает различные параметры, такие как количество поисков, отправленных сообщений, количество неудачных поисков, задержка сообщений и количество переходов. На этом же уровне P2P-приложения, например, DHT TestApp, используются для тестирования DHT-приложения путем выполнения

DHT-запросов PUT и GET со случайно сгенерированными элементами данных. Эти приложения реализованы в виде модулей и взаимодействуют с оверлеями через KBR API, которые представляют базовые возможности, общие для всех структурных оверлеев. На уровне оверлея OverSim предоставляет ряд различных сетевых моделей. В оверлейной сети структура INET может быть использована для реализации протокола управления сообщениями, а также поддерживает моделирование беспроводных и мобильных сетей. Кроме того, OverSim предоставляет несколько моделей оверлея для детального моделирования, а также предоставляет оверлей с одним узлом для моделирования одного УМ.

В этой симуляции рассчитывается и регистрируется эффективность управления сообщениями, количество переходов при поиске и задержка успешного поиска для УМ. Полученные данные являются средним значением трех независимых экспериментов. Для измерения производительности были проанализированы следующие показатели:

- эффективность управления сообщениями, также известная как коэффициент успешности поиска, рассчитывается путем взятия отношение общего количества сообщений поиска, которые были успешными до истечения времени ожидания, к общему количеству сообщений поиска, переданных системой. Эта метрика рассчитывается следующим образом:

$$\text{Эффективность управления сообщениями} = \frac{\text{Успешный поиск}}{\text{Полный поиск}} \times 100\% \quad (20)$$

- количество шагов или рабочая нагрузка на один узел рассчитывается путем соотношения общего количества широковещательных сообщений и общего количества узлов, необходимых для успешной пересылки сообщений на узел назначения. Вычисляется по формуле:

$$\text{Количество переходов} = \frac{\text{общее количество широковещательных сообщений}}{\text{общее количество узлов, необходимых для пересылки сообщений}} \quad (21)$$

- задержка успешного поиска определяется как время, необходимое пакету данных, чтобы добраться от одной назначенной точки к другой, а иногда измеряется как время, необходимое для возвращения пакета отправителю.

В этом исследовании моделируется производительность управления сообщениями для различных размеров сети и среднего времени жизни УМ, как указано в таблице 1. Количество УМ было ограничено 500. Среднее время жизни узлов варьируется от 100 до 1000 секунд. Это ограничение связано с ограниченными ресурсами для выполнения моделирования.

Таблица 3.1 – Параметры, используемые для моделирования

Название параметра	Архитектура сети		
	Chord	Pastry	Kademlia
Количество узлов	100,200,300,400, 500	100,200,300,400, 500	100,200,300,400, 500
Средняя продолжительность жизни узла (сек.)	100,300,500,700, 1000	100,300,500,700, 1000	100,300,500,700, 1000

Для сбора результатов эффективности управления сообщениями было проведено два сценария. Первый эксперимент - увеличение размера сети от 100 до 500 узлов с фиксированным средним временем жизни 1000 сек. Рисунок 3.5 иллюстрирует эффективность управления сообщениями в зависимости от размера сети.

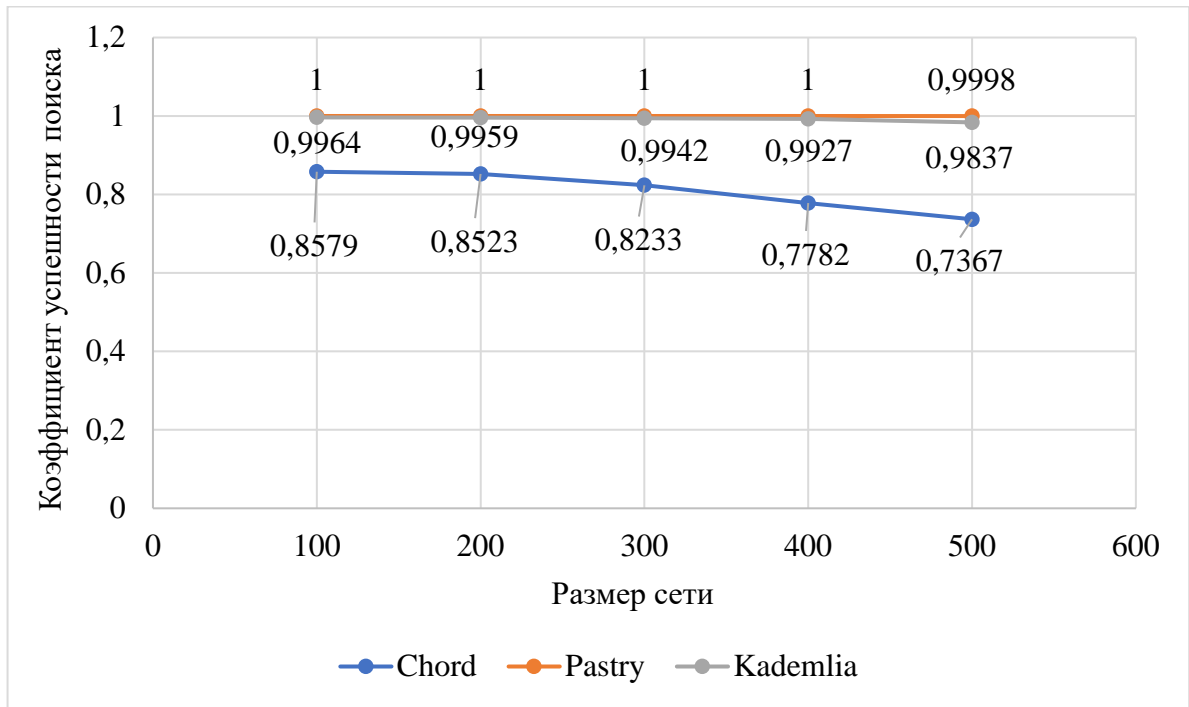


Рисунок 3.5 – Результаты для Chord, Pastry и Kademlia по эффективности управления сообщениями в зависимости от размера сети

У Chord самая низкая эффективность управления сообщениями по размеру сети по сравнению с Pastry и Kademlia. Самый низкий коэффициент эффективности управления сообщениями у Chord, где он составляет 0,8579 при 100 узлах и постепенно уменьшается при увеличении размера сети до 500 узлов. Pastry поддерживает эффективность управления сообщениями на уровне 1,0 до размера сети 500 узлов, затем она снижается до 0,9998. У Kademlia этот показатель немного ниже, чем у Pastry. Если эффективность управления сообщениями равна 1,0, что означает хорошее управление сообщениями с высокой стабильностью сети. Chord организует свою таблицу маршрутизации в одномерном пространстве, в то время как Pastry и Kademlia имеют большую таблицу маршрутизации и организуют свою таблицу маршрутизации в двухмерном пространстве.

Вторым сценарием эффективности управления сообщениями является увеличение среднего времени жизни при фиксированном размере сети (100 узлов). Рисунок 3.6 показывает зависимость эффективности управления сообщениями от среднего времени жизни.

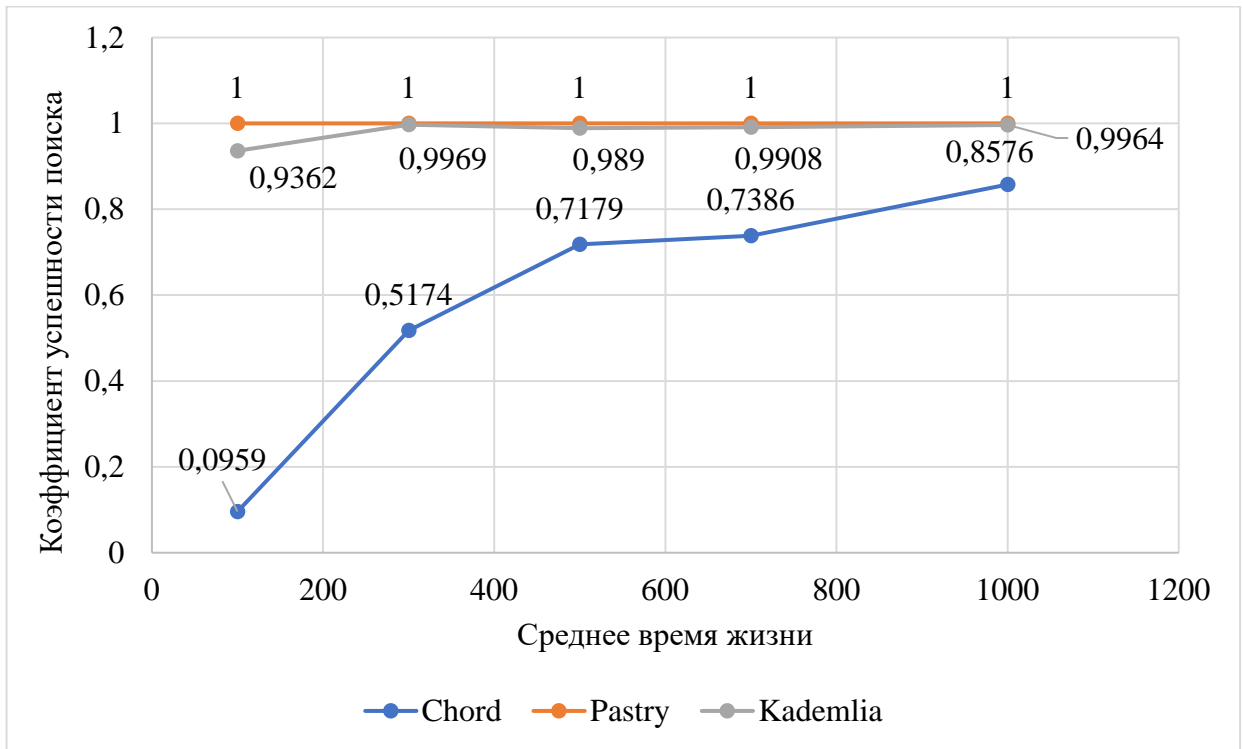


Рисунок 3.6 – Результаты для Chord, Pastry и Kademia по эффективности управления сообщениями в сравнении со средним временем жизни

На рисунке показаны схожие результаты, где Chord имеет наименьшее значение по сравнению с алгоритмом Pastry и Kademia. Отношение составляет 0,0959 (алгоритм Chord) на 100 секунд времени жизни. Отношение в Chord линейно увеличивается по мере увеличения среднего времени жизни. У Kademia этот показатель немного снижается по сравнению с Pastry. Самая высокая эффективность управления сообщениями - коэффициент 1.0, который имеет Pastry при среднем времени жизни от 100 секунд до 1000 секунд. Это доказывает, что Pastry - лучший алгоритм для эффективности управления сообщениями.

Для расчета количества хопов при поиске были проведены аналогичные сценарии. Сначала увеличивался размер сети от 100 до 500 узлов с фиксированным средним временем жизни 1000 секунд. Рисунок 3.7 иллюстрирует зависимость количества просмотренных хопов от размера сети.

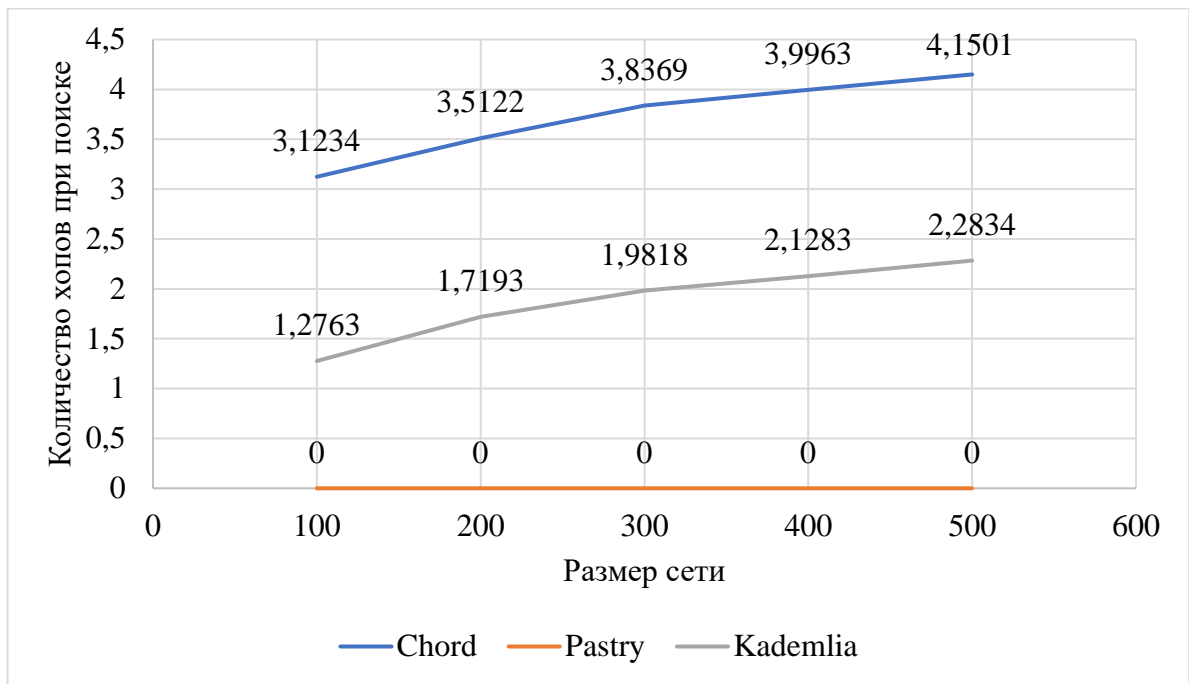


Рисунок 3.7 – Результаты для Chord, Pastry и Kademlia по количеству переходов при поиске в зависимости от размера сети

Из рисунка видно, что количество скачков при поиске для Pastry показывает постоянное соотношение 0, где значения были слишком маленькими. Это связано с процессом округления значений. Между Kademlia и Chord, первый имеет лучший показатель количества хопов при 100 узлах, где количество хопов составляет 1,2763. У Chord этот показатель равен 3,1234. Когда сеть увеличивается до 500 узлов, среднее количество поисковых хопов для Kademlia и Chord увеличивается соответственно. Результат показывает, что Pastry имеет наименьшее количество поисковых переходов по сравнению с Chord и Pastry. Чем меньше число поисковых переходов, тем лучше протокол управления сообщениями. Этот результат повторяет предыдущий результат эффективности управления сообщениями, где Pastry имеет коэффициент эффективности управления сообщениями 1.0.

Второе моделирование - увеличение среднего времени жизни от 100 до 1000 секунд при фиксированном размере сети в 100 узлов. Рисунок 3.8 показывает зависимость количества хопов при поиске от среднего времени жизни.

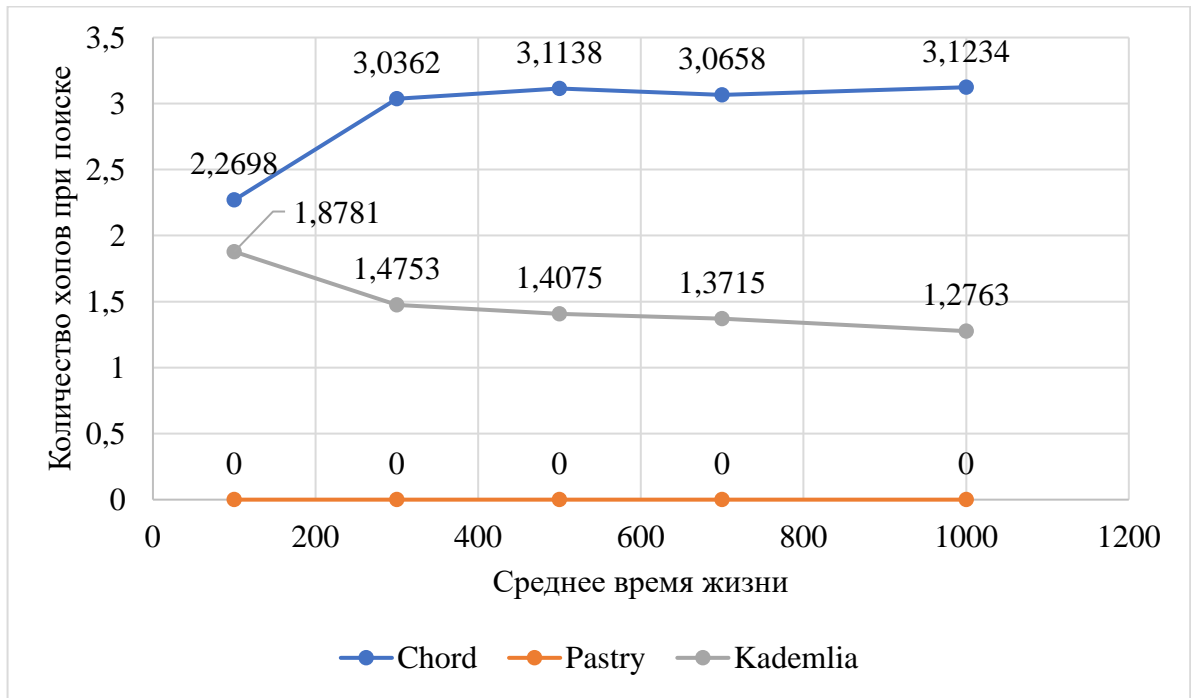


Рисунок 3.8 – Результаты для Chord, pastry и kademia, основанные на соотношении количества хопов при поиске и среднего времени жизни

Из рисунка видно, что Pastry имеет схожие результаты в сравнении размера сети с количеством хопов при поиске, где количество переходов при поиске близко к нулю. Chord имеет наибольшее количество хопов при поиске. Таким образом, Pastry демонстрирует лучшее управление сообщениями по количеству просмотренных хопов среди других алгоритмов.

Аналогично задержке успешного поиска, были проведены два сценария с увеличением размера сети от 100 узлов до 500 узлов с фиксированным средним временем жизни 1000 секунд. На рисунке 3.9 показан график зависимости времени успешного поиска от размера сети.

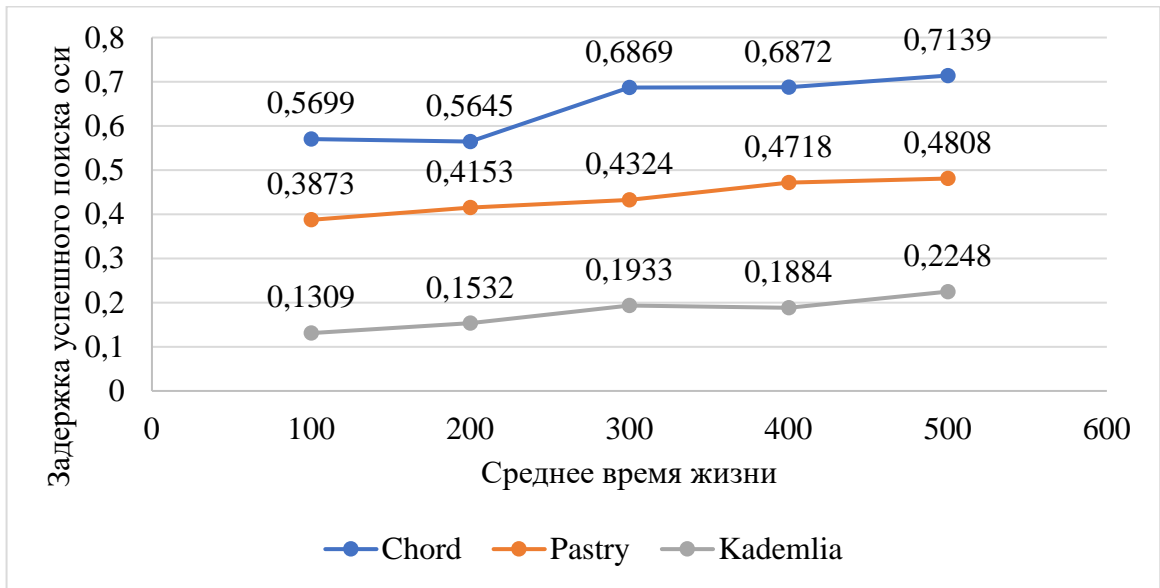


Рисунок 3.9 – Результаты для Chord, Pastry и Kademlia по задержке успешного поиска в зависимости от размера сети

Задержка успешного поиска — это время задержки, которое требуется для хранения или извлечения пакетов данных. Чем меньше задержка успешного поиска, тем выше производительность управления сообщениями. Как показано на рисунке, Chord имеет самую высокую задержку успешного поиска, а Kademlia - самую низкую. Второе моделирование показало увеличение среднего времени жизни при фиксированном размере сети в 100 УМ. Рисунок 3.10 иллюстрирует зависимость времени успешного поиска от среднего времени жизни. Chord имеет самую высокую задержку успешного поиска по сравнению с Pastry и Kademlia. У Kademlia самая низкая задержка успешного поиска.

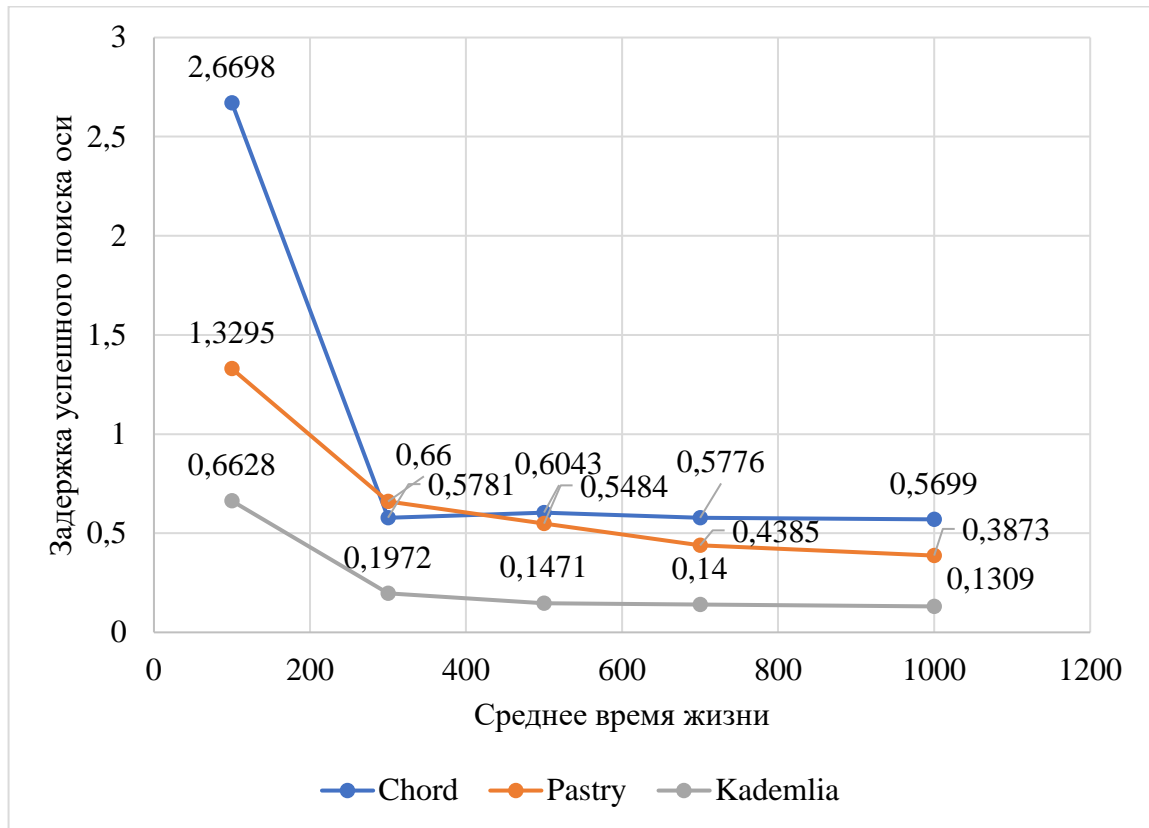


Рисунок 3.10 – Результаты по Chord, Pastry и Kademia на основе зависимости времени успешного поиска от среднего времени жизни

Результат проведенного исследования показывает, что Pastry имеет самый высокий коэффициент эффективности управления сообщениями. Chord имеет самую низкую эффективность управления сообщениями из-за того, что Chord организует свою таблицу маршрутизации в одномерном пространстве, в то время как Pastry и Kademia имеют большую таблицу маршрутизации и организуют свою таблицу маршрутизации в двухмерном пространстве. Количество хопов определяется как объем работы на один УМ в сети. Низкий коэффициент количества переходов при поиске хорош для управления сообщениями, когда нагрузка на один узел также низка. Результаты по количеству просмотренных хопов показывают, что Pastry имеет самый низкий коэффициент, за ним следуют Kademia и Chord. Значение, полученное для Pastry в результате моделирования, округляется до 0. Последнее моделирование показывает, что Kademia получила самый низкий коэффициент задержки успешного поиска, за ней следуют Pastry и

Chord. Задержка успешного поиска между Kademlia и Pastry также не имеет существенных различий.

3.2.2.2 Разработка модификации алгоритма наложенного управления сообщениями Pastry для наложенного управления многоадресной рассылкой подмножества ИАМ

Исходя из результатов исследования, для решения проблем, связанных, как с масштабированием пиринговой системы, что является особенно важным критерием для крупномасштабных сетей с IoT устройствами, производительностью, так и с ее устойчивостью к сбоям/отказам отдельных узлов сети в качестве протокола децентрализованного взаимодействия всех узлов СМ был выбран протокол Pastry.

В обобщенном виде процесс наложенного управления сообщениями Pastry представлен на рисунке 3.11.

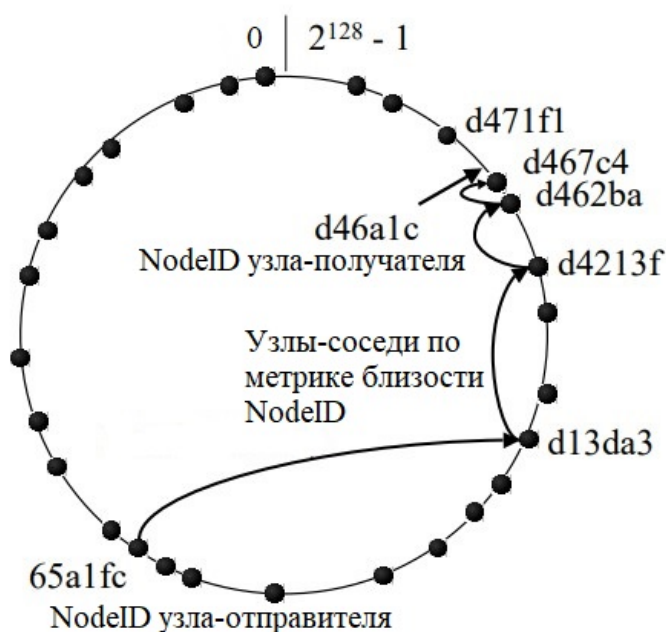


Рисунок 3.11 – Процесс наложенного управления сообщениями Pastry

Предлагаемая схема модификации алгоритма наложенного управления сообщениями Pastry для случая, когда сообщение с ключом d формируется узлом с $NodeID = a$ представлена на рисунке 3.12.

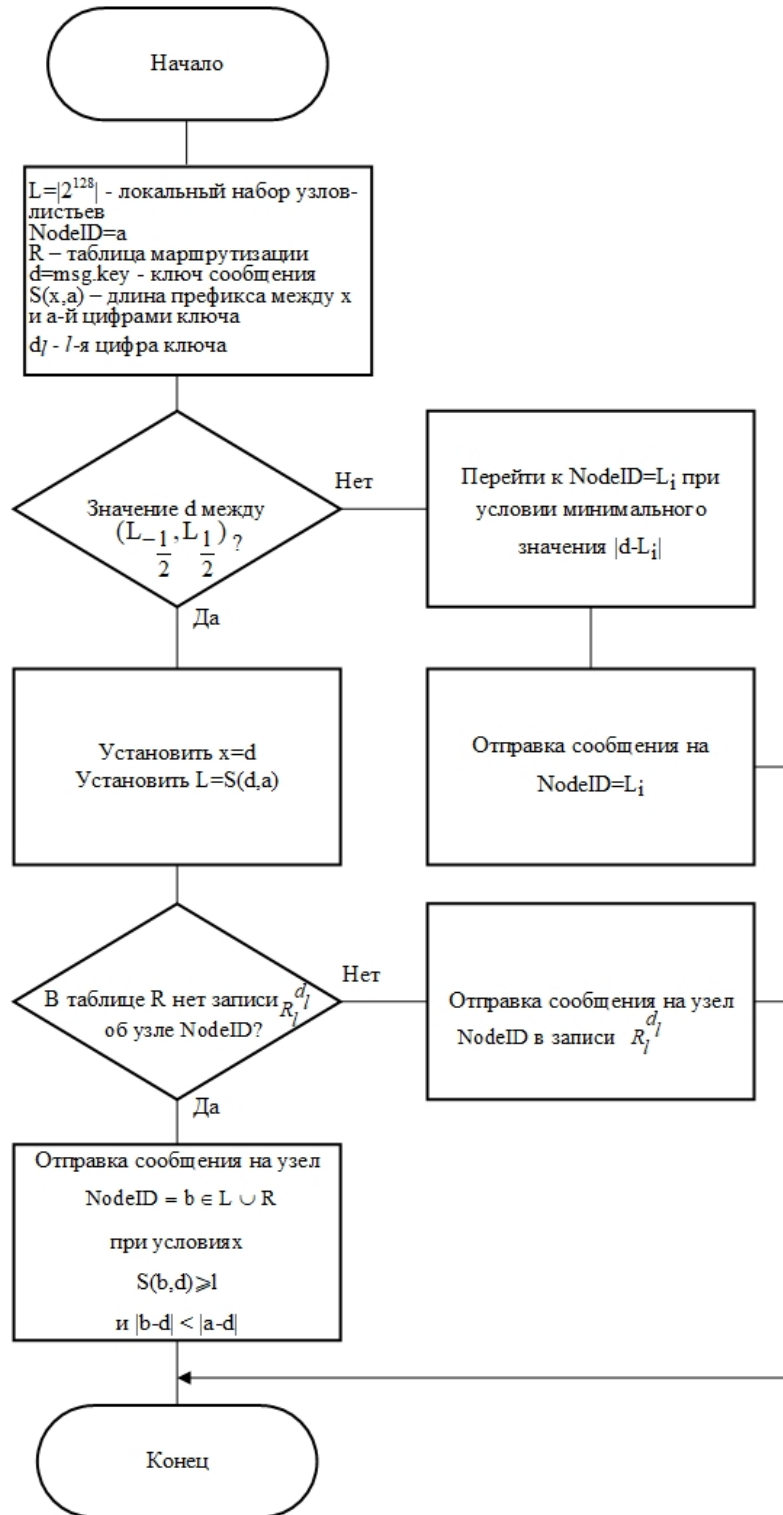


Рисунок 3.12 – Схема алгоритма процедуры наложенного управления сообщениями на основе Pastry

Как было указано выше, в предельном случае количество шагов для множества K ИАМ_{УМК} равно $\log_{2b} K$. При этом на практике реальное число шагов существенно зависит от вероятности того, что ключ сообщения d уже содержит префикс NodeID узла-отправителя и промежуточных узлов-соседей.

Поскольку, исходя из представленного процесса наложенного управления сообщениями на основе Pastry (рисунок 3.12), передача сообщения может быть выполнена с использованием одной из трех процедур: m_A - по множеству L узлов-листьев (leaf set) (блок 5 рисунка 3.12), m_B - с использованием таблицы маршрутизации R (блок 7 рисунка 3.12), или m_C - с использованием узла, принадлежащего множеству $L \cup R$ (блок 8 рисунка 3.12), предлагается определить вероятность перехода к любой из процедур, как P_x , где $x \in \{A, B, C\}$ при шаге $h+1$ для передачи сообщения с произвольным ключом, начиная от случайно выбранного узла из множества K , с подмножеством узлов-листьев l , как $P(h, l, K, m_x)$. На основе этого определим вероятности $P(h, l, K, m'_A)$, $P(h, l, K, m'_B)$, $P(h, l, K, m'_C)$, как вероятности того, что алгоритм, выбрав соответствующую процедуру, завершит работу на шаге $h+1$, обнаружив соседний узел с метрикой, численно близкой метрике NodeID. Далее введем вероятность полной группы событий $P(h, l, K, m_x)$ для $h \in \left\{0, \frac{128}{b-1}\right\}$ и $x \in \{A, A', B, B', C, C'\}$, и запишем ее в виде матрицы вероятностей:

$$\begin{pmatrix} P(h, l, N, m_A) \\ P(h, l, N, m'_A) \\ P(h, l, N, m_B) \\ P(h, l, N, m'_B) \\ P(h, l, N, m_C) \\ P(h, l, N, m'_C) \end{pmatrix} = \sum_{d=0}^{2^b-1} \sum_{n_0=0}^N b(n_0, N, \frac{1}{2^b}) \quad (23)$$

Введение матрицы вероятностей (выражение 23) для предлагаемого алгоритма наложенного управления сообщениями на основе модификации алгоритма Pastry предоставляет приоритетный выбор одной из трех процедур

управления, что обеспечивает сокращение времени, затрачиваемого на управление сообщением (передачу) между множеством узлов дерева многоадресной рассылки для сформированной темы, связанной с конкретным моментом возникновения нестационарной нагрузки на корневом узле дерева рассылки.

Таким образом совокупность предлагаемых алгоритмов (рисунки 3.4 и 3.12) позволяет решить задачу сокращения времени задержки передачи данных о моментах возникновения нестационарной нагрузки соседним ИАМ_{УМК}, расположение которых соответствует направлению распространения временного лага нагрузки.

3.3 Выводы по главе

1. Разработан обобщенный алгоритм динамической настройки параметров разработанной нейросетевой модели на основе сети LSTM для достижения удовлетворительной точности прогнозирования при ее многократной реконфигурации, позволяющий свести к минимуму общие временные затраты на обучение нейросетевой модели, что является критичным для условий распространения нестационарной нагрузки в узлах мониторинга.

2. Разработан обобщенный алгоритм взаимного информационного согласования ИАМ СМ, состоящий из двух взаимосвязанных процедур: алгоритма управления многоадресной рассылкой сообщений (1) и алгоритма наложенного управления сообщениями (2).

Для (1) разработаны алгоритмы по управлению передачей и приемом сообщений темы многоадресной рассылки. Для (2) обосновано выбран протокол децентрализованного взаимодействия ИАМ_{УМК} и модифицирован алгоритм наложенного управления сообщениями Pastry.

ГЛАВА 4. РАЗРАБОТКА АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОЙ МНОГОАГЕНТНОЙ СИСТЕМЫ МОНИТОРИНГА ОБЪЕКТОВ МУЛЬТИСЕРВИСНОЙ СЕТИ СВЯЗИ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ МОНИТОРИНГА

В главе подробно описана разработанная двухуровневая архитектура распределенной многоагентной системы мониторинга объектов мониторинга на примере мультисервисной сети связи.

Разработано программное обеспечение интеллектуальных агентов мониторинга, предназначенных для функционирования на структурных компонентах МСС в качестве альтернативы датчиков централизованных систем мониторинга типа wiSLA. Приведена их логическая структура и показан вариант программно-аппаратной реализации.

В качестве оценки эффективности предлагаемых решений представлены результаты сравнительного численного эксперимента.

4.1 Разработка архитектуры распределенной многоагентной системы мониторинга объектов мультисервисной сети связи

Как сказано в главе 1 в п. 1.1 объемы трафика с каждым годом стремительно увеличиваются, а количество устройств различной функциональности в сетях растет. Это напрямую влияет на нагрузку сетевых каналов и требует развития существующих СМ для предоставления необходимого уровня обслуживания.

Основой развития СМ МСС является модификация существующих, либо создание новых архитектурных решений. Задачи, решаемые в настоящем исследовании, направлены на разработку архитектуры распределенной многоагентной СМ, вариант которой, в отличие от СМ централизованного типа, таких, как wiSLA формирует для ОМ МСС децентрализованную структуру, обеспечивающую эффективное управление УМ МСС в кратковременные периоды возникновения нестационарной нагрузки на них, благодаря учету временного лага

распространения этой нагрузки по множеству УМ (глава 1 в п. 1.2). Поскольку программно-аппаратная реализация функциональности ИАМ существенно зависит от особенностей ОМ. В качестве примера, разработка их архитектурных решений будет вестись для объектов транспортной подсистемы МСС, которые образуют множество УМ.

Как было указано выше, разрабатываемая архитектура распределенной многоагентной СМ предлагается в качестве альтернативного решения существующим СМ МСС с централизованной архитектурой на основе wiSLA, которая представлена на рисунке 4.1.

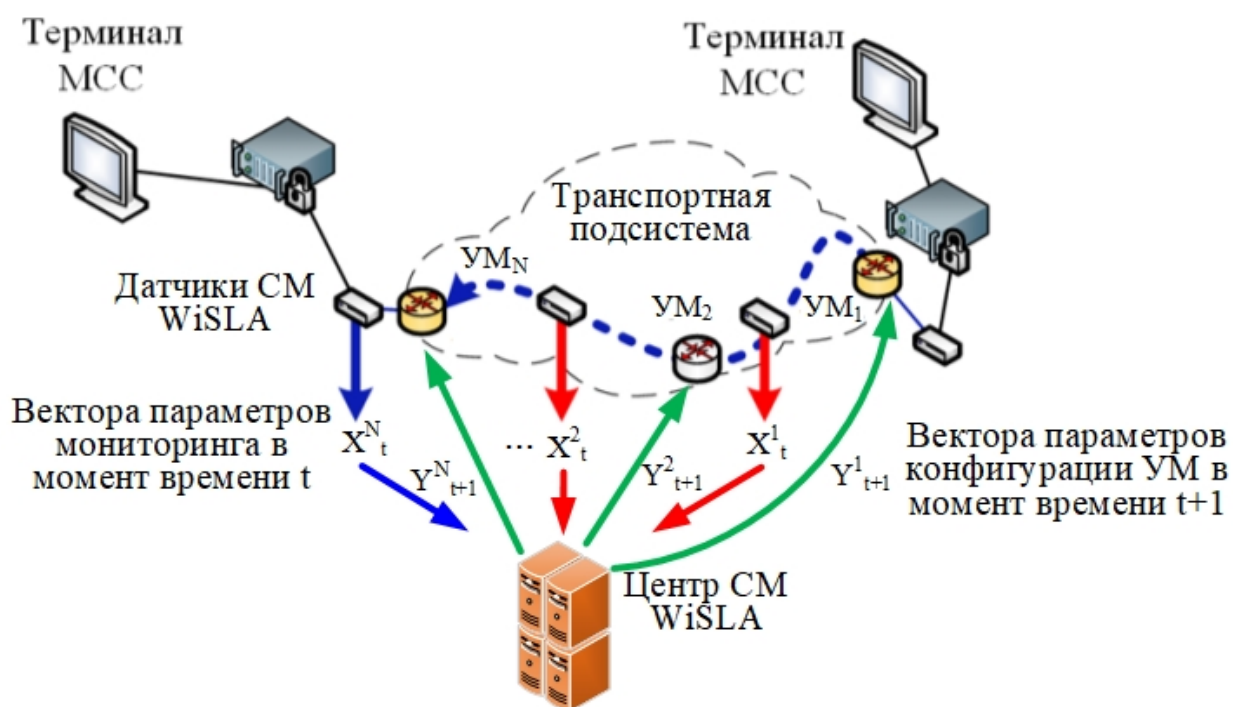


Рисунок 4.1 – Централизованная система мониторинга МСС на базе WiSLA

На основе разработанного в главе 2 математического обеспечения ИАМ, а также разработанных в главе 3 алгоритмов динамического дообучения нейросетевой модели и наложенного управления многоадресной рассылкой сообщений ИАМ, архитектуру распределенной многоагентной СМ предлагается реализовать в двухуровневом варианте восходящей схемы уровней.

1. Нижележащий уровень наложенного управления сообщениями Pastry, обеспечивающий формирование оверлейной сети ИАМ для решения задачи доставки сообщений с данными параметров мониторинга УМ и их прогнозных значений.

2. Вышележащий уровень нейросетевой модели ИАМ, обеспечивающий решение задачи получения такой конфигурации LSTM-сети, которая максимизирует точность прогнозирования параметров мониторинга в моменты времени $t+p$, где p – глубина прогнозирования. В обобщенном виде предлагаемая архитектура представлена на рисунке 4.2.

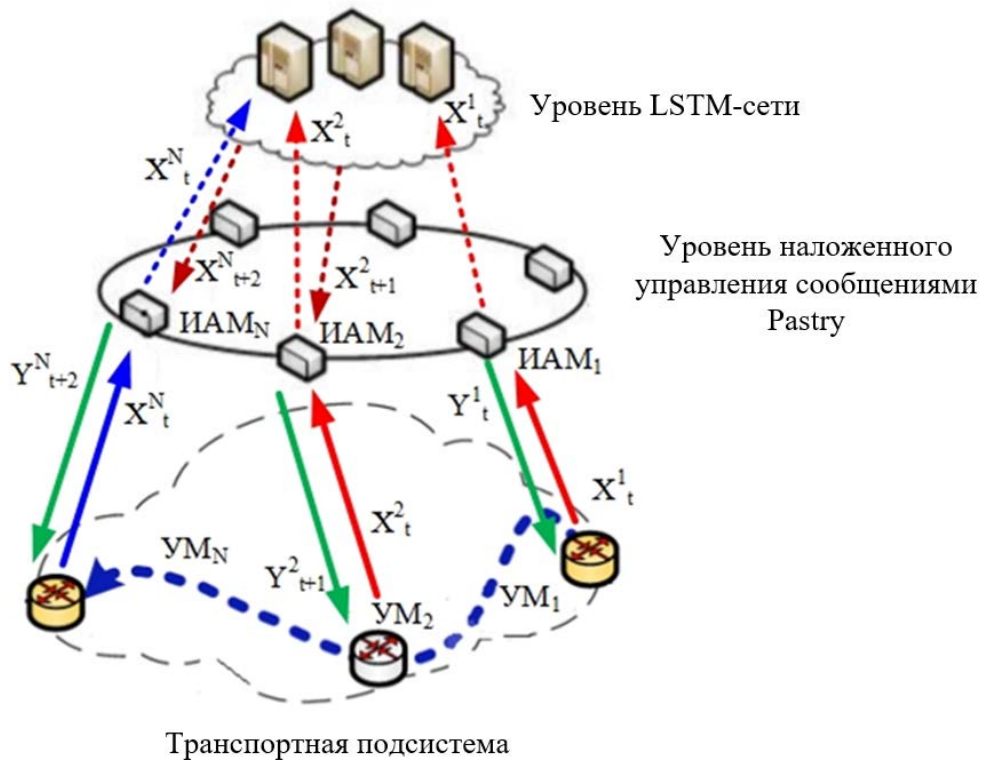


Рисунок 4.2 – Двухуровневая архитектура распределенной многоагентной системы мониторинга

На рисунке показана структура сети - ОМ, на который оверлейно наложена разработанная распределенная многоагентная СМ с двухуровневой архитектурой, ИАМ которой получают параметры нагрузки X_t^N от закрепленных за ними УМ,

обмениваются информацией между собой и анализируют данные как от закрепленного УМ, так и от соседних, вырабатывая управляющее воздействие Y_t^N на закрепленный УМ [77,78]. Более подробно процессы уровня LSTM-сети и уровня наложенного управления сообщениями Pastry представлены в главах 2 и 3 соответственно. Предложенные архитектурные решения требуют разработки соответствующего программного обеспечения их компонентов.

4.2 Разработка программного обеспечения интеллектуальных агентов мониторинга

В рамках разработки аппаратной вычислительной платформы ИАМ необходимо учитывать его функциональные возможности. Множество ИАМ в составе СМ МСС предназначены для анализа трафика в реальном времени и прогнозирования критических моментов (пиковых нагрузок) в определенный момент времени. С технической точки зрения, ИАМ является УМ или канал сети за которым закреплен встроенный компьютер с невысокой вычислительной мощностью. Поскольку, для разрабатываемой СМ была выбрана парадигма полной децентрализации с распределенной обработкой данных, что подразумевает отсутствие единого ЦОД, предполагается, что небольшая вычислительная мощность каждого ИАМ в совокупности с протоколом децентрализованного взаимодействия всех ИАМ СМ даст необходимую вычислительную мощность, сопоставимую с мощностью ЦОД. На рисунке 4.3 представлена логическая структура ИАМ.

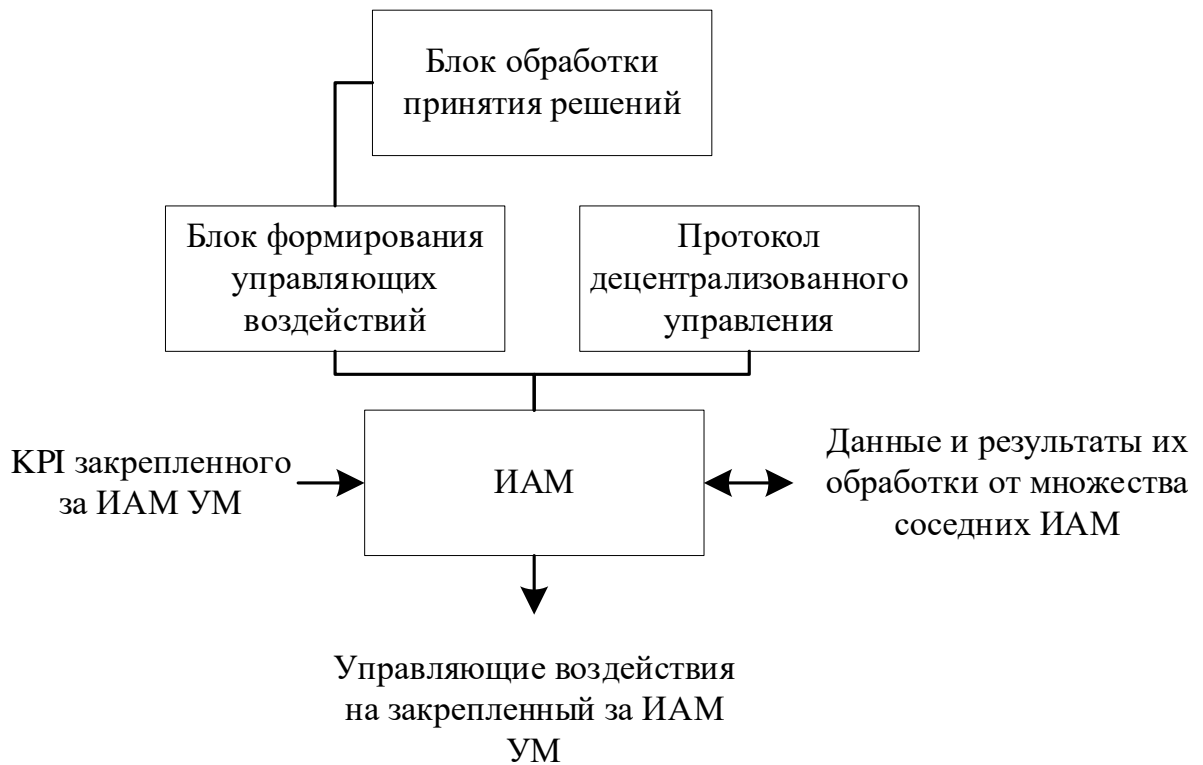


Рисунок 4.3 – Логическая структура ИАМ

Из рисунка видно, что каждый ИАМ СМ получает КРІ от закрепленного за ним УМ или канала в сети и обрабатывает полученную информацию. Для более точного принятия управляющего воздействия (блок формирования управляющих воздействий) на УМ ИАМ СМ под управлением протокола децентрализованного взаимодействия обмениваются между собой информацией и результатами их обработки для уточнения общего состояния ОМ [84].

Обработка данных происходит за счет применения нейронной сети (блок обработки принятия решения), которая заранее обучается на множестве факторов, получаемых из поступающих КРІ, что существенно снижает вычислительную нагрузку на УМ. Прогнозирующая модель, закладываемая в разрабатываемую систему проактивного мониторинга, в качестве независимой переменной использует время, а результатом модели является прогнозируемое значение в определенный момент времени. Каждый ИАМ в реальном времени собирает множество факторов, получаемых из КРІ, образующие на временном интервале

последовательность значений, которая, в свою очередь, представляет многомерный временной ряд.

Для реализации прототипа ИАМ предлагается использовать комбинированную аппаратную вычислительную платформу, состоящую из следующих вычислительных модулей:

- одноплатная вычислительная платформа «Raspberry Pi 3 Model B+» для предоставления ресурсов программному обеспечению наложенного управления сообщениями Pastry и модулям расчета вектора параметров конфигурации подконтрольного УМ (рисунок 4.4);



Рисунок 4.4 – Одноплатная вычислительная платформа «Raspberry Pi 3 Model B+»

- вычислительный модуль «Intel NCS2» на базе нейропроцессора Intel Movidius Myriad X Vision Processing Unit для решения задач обучения и функционирования LSTM-сети (рисунок 4.5).



Рисунок 4.5 – Вычислительный модуль «Intel NCS2»

Для сбора данных KPI с закрепленного за ИАМ УМ, в качестве примера, предлагается использовать технологию активного мониторинга трафика - Cisco IP SLA. На рисунке 4.6 показана техническая реализация ИАМ СМ.

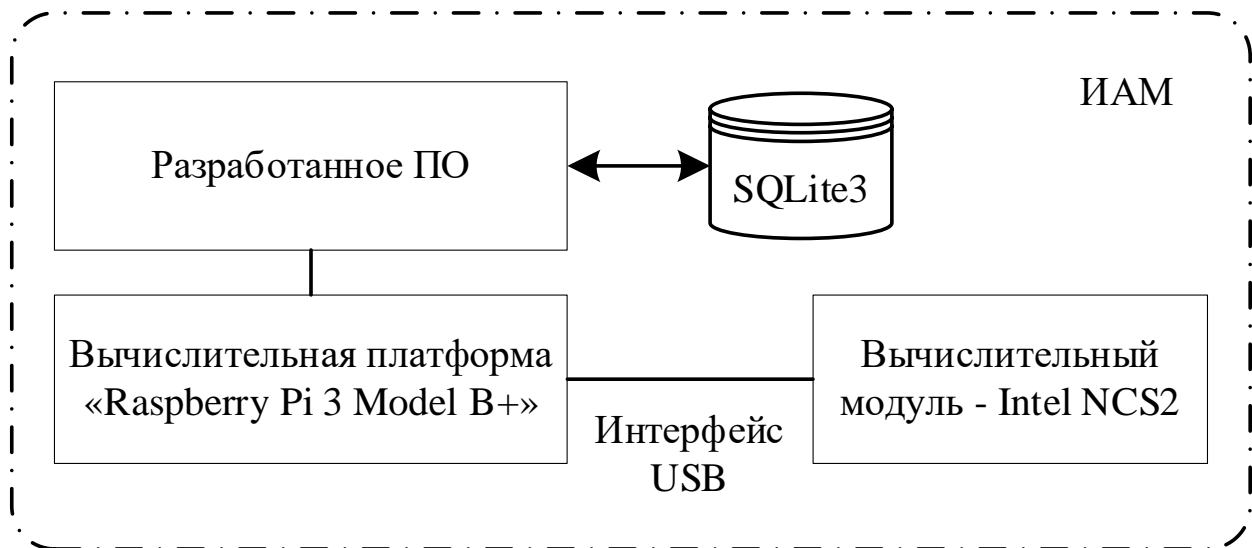


Рисунок 4.6 – Техническая реализация ИАМ СМ

Как видно из рисунка, на основе комбинированной аппаратной вычислительной платформы в ИАМ функционирует специальное разработанное ПО, обеспечивающее необходимый функционал по приему, обработки, отправки данных, а также по формированию управляющих воздействий на закрепленный УМ. ПО было разработано посредством использования языка программирования python и его штатных библиотек и фреймворков, таких как: Keras, Stasmodels и TensorFlow [79,80,81]. В качестве базы данных (БД) выбрана компактная встраиваемая система управления базами данных (СУБД) SQLite3, ведущей обработку информационно-логической модели БД из четырех таблиц: две - для обработанных и полученных данных, таблица допустимых значений нагрузки на УМ, а также таблица с параметрами обученных моделей LSTM-сети.

Все данные и результаты их обработки, получаемые ИАМ от соседних ИАМ, а также собранные KPI от закрепленного за ним УМ, посредством применения технологии Cisco IP SLA с помощью разработанного ПО записываются в базу данных, хранение которой организовано на запоминающем устройстве

вычислительной платформы «Raspberry Pi 3 Model B+». После чего, поступившие данные обрабатываются предобученными моделями LSTM-сети посредством использования вычислительного модуля Intel NCS2, подключенного по порту USB к «Raspberry Pi 3 Model B+», для ускорения работы нейронной сети. Результаты обработки записываются в БД, отправляются соседним ИАМ и сравниваются с допустимыми значениями нагрузки на УМ. В случае, превышения таких значений формируются управляющие воздействия на закрепленный УМ, например, происходит изменение таблицы маршрутизации для поддержания равновесного состояния нагрузки в сети, иначе корректирующих решений не выдаётся.

4.3 Проведение численного эксперимента для оценки эффективности полученных решений

Обобщенная схема работы разработанной распределенной многоагентной децентрализованной СМ МСС показана на рисунке 4.7.

С целью оценки эффективности полученных решений необходимо провести соответствующий численный эксперимент. Так как для проведения эксперимента на реально существующей МСС, необходима крупномасштабная сеть с большим количеством узлов и большим трафиком, было принято решение создать комбинированную имитационную модель.

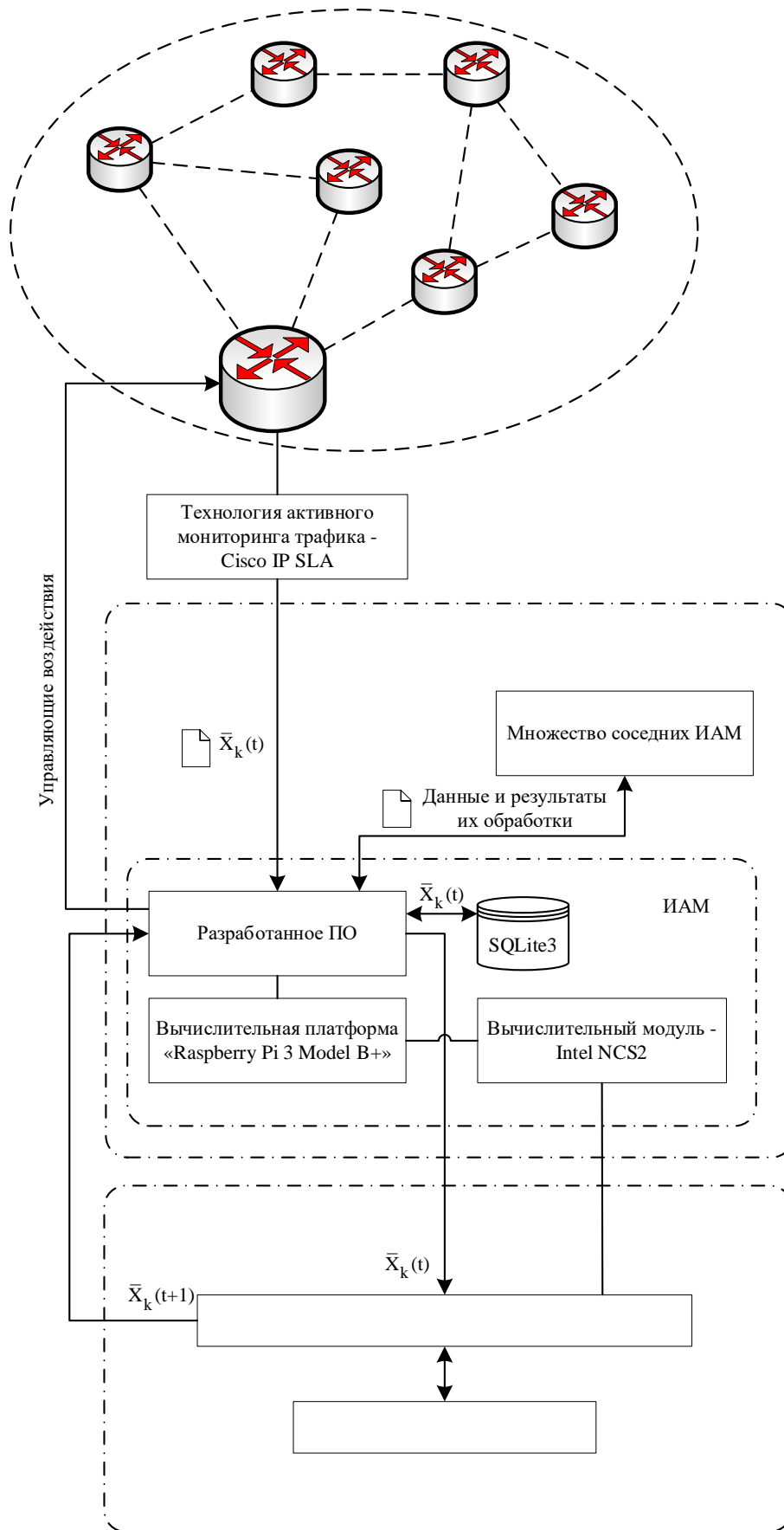


Рисунок 4.7 – Обобщенная схема работы разработанной распределенной многоагентной децентрализованной CM MCC

Для проведения сравнительного эксперимента на структурах МСС разного масштаба предлагается реализовать два варианта фрагмента МСС с 10 и 30 УМ соответственно. Таким образом определим:

- объекты мониторинга:

$$OM_{10} = \{UM_1, UM_2, \dots, UM_{10}\} \text{ и } OM_{30} = \{UM_1, UM_2, \dots, UM_{30}\}.$$

- системы мониторинга:

$$CM_{ц} = \{MM_{OM}, \{AM_{UM_1}, \dots, AM_{UM_{10}}\}\} - \text{централизованная CM}$$

типа wiSLA;

$$CM_{дц} = \{IAM_{UM_1}, \dots, IAM_{UM_{10}}\} - \text{предлагаемый вариант CM с}$$

децентрализованной структурой и рассмотренной в п. 4.2 двухуровневой архитектурой.

Для моделирования структуры ОМ и генерации моментов возникновения нестационарной нагрузки на множестве УМ предлагается использовать фреймворк Cisco Packet Tracer [82]. На рисунках 4.8 и 4.9 показаны сгенерированные в Cisco Packet Tracer структуры МСС на основе которых организуется сравнительный эксперимент.

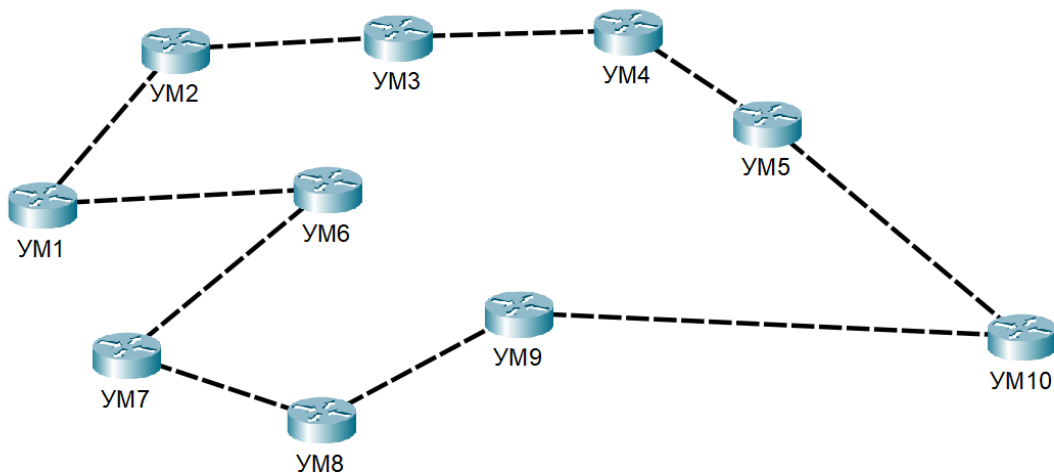


Рисунок 4.8 – Структура сети, состоящая из 10 УМ

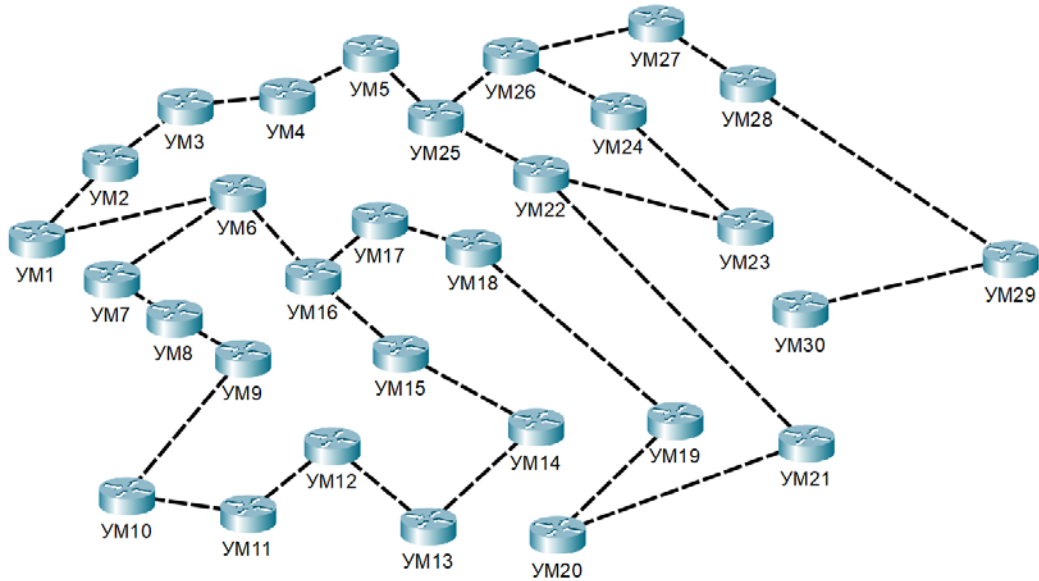


Рисунок 4.9 – Структура сети, состоящая из 30 УМ

Для построения модели разработанной распределенной многоагентной СМ, а также алгоритма наложенного управления сообщениями и многоадресной рассылки сообщений между множеством моделируемых ИАМ будем использовать среду имитационного моделирования AnyLogic [83]. В работах [85,86] процесс управления сообщениями в МСС моделируется на основе методологии систем массового обслуживания (СМО), поэтому построение модели такой системы выполняется с помощью элементов «библиотеки моделирования процессов» AnyLogic.

Источником заявок являются выбранные УМ в сети с различной интенсивностью поступления (моменты пиковых нагрузок). Очередь моделирует маршрутизатор с буфером вместимость которого равна 11 с закрепленным за ним ИАМ. Задержка моделирует временной лаг распространения нагрузки и формируется согласно закону нормального распределения.

Для СМц на базе wiSLA эксперимент проводился на экспериментальном участке реальной МСС. Для СМдц был проведен имитационный эксперимент. Расчет скважности в СМц происходит по следующей формуле:

$$S_k^j = t_{delay}^j = t_{сб}^j + t_{np}^j + t_{обр}^j + t_{nep}^j + t_{конф}^j;$$

Для СМдц: $S_k^j = t_{delay}^j = t_{обуч.LSTM}^j + t_{вз.согл.}^j$; То есть необходимо собрать статистику по всем временам каждого УМ и взять их среднее значение.

На рисунке 4.10 показана имитационная модель для OM_{10} , построенная в AnyLogic.

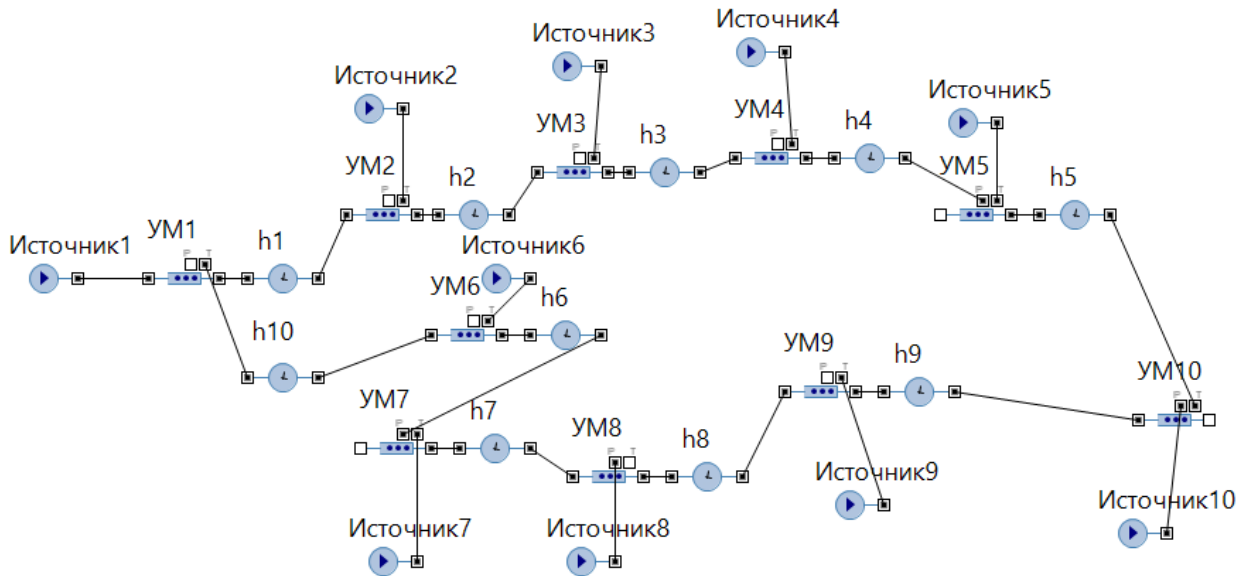


Рисунок 4.10 – Имитационная модель для OM_{10}

В таблице 4.1 представлены обобщенные результаты наблюдений для OM_{10} .

Таблица 4.1 - Результаты наблюдений для OM_{10}

Количество моментов нестационарной нагрузки в произвольные моменты времени t	Скважность S_k^j , секунд	
	СМц	СМдц
10	3,5	1,5
30	4,5	2,5
70	7	5,5
100	9,7	8,5
150	30	12
200	37	22

На рисунке 4.11 представлен сравнительный график, показывающий результаты наблюдений для OM_{10} .

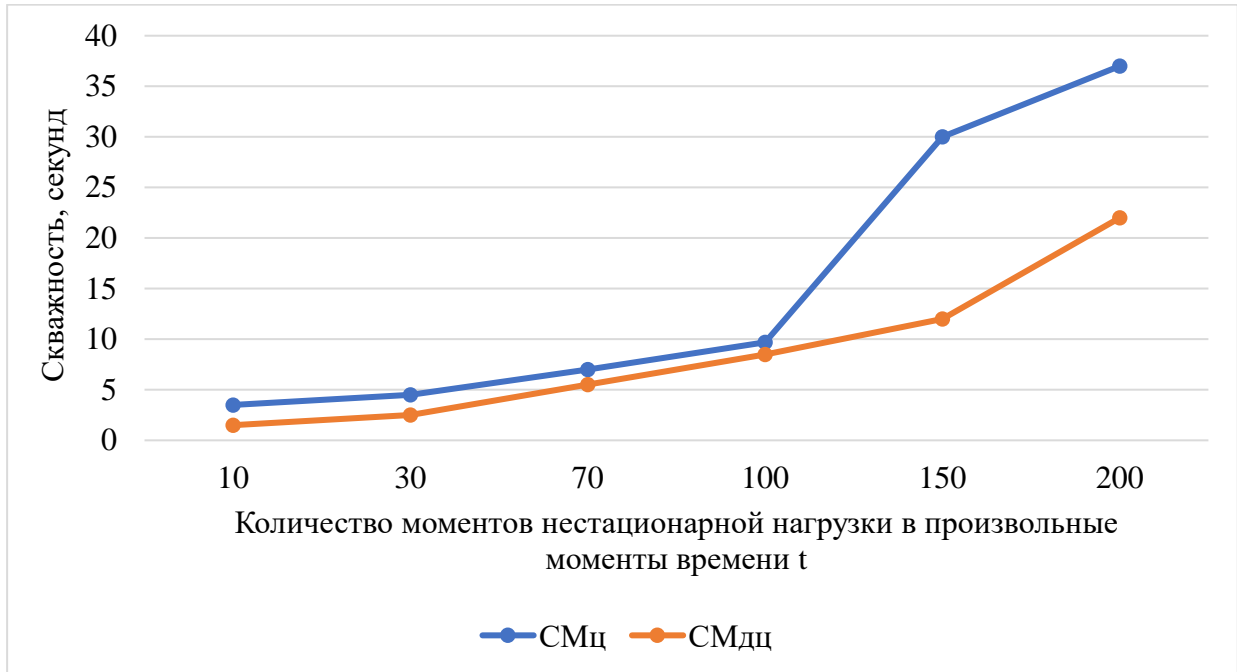
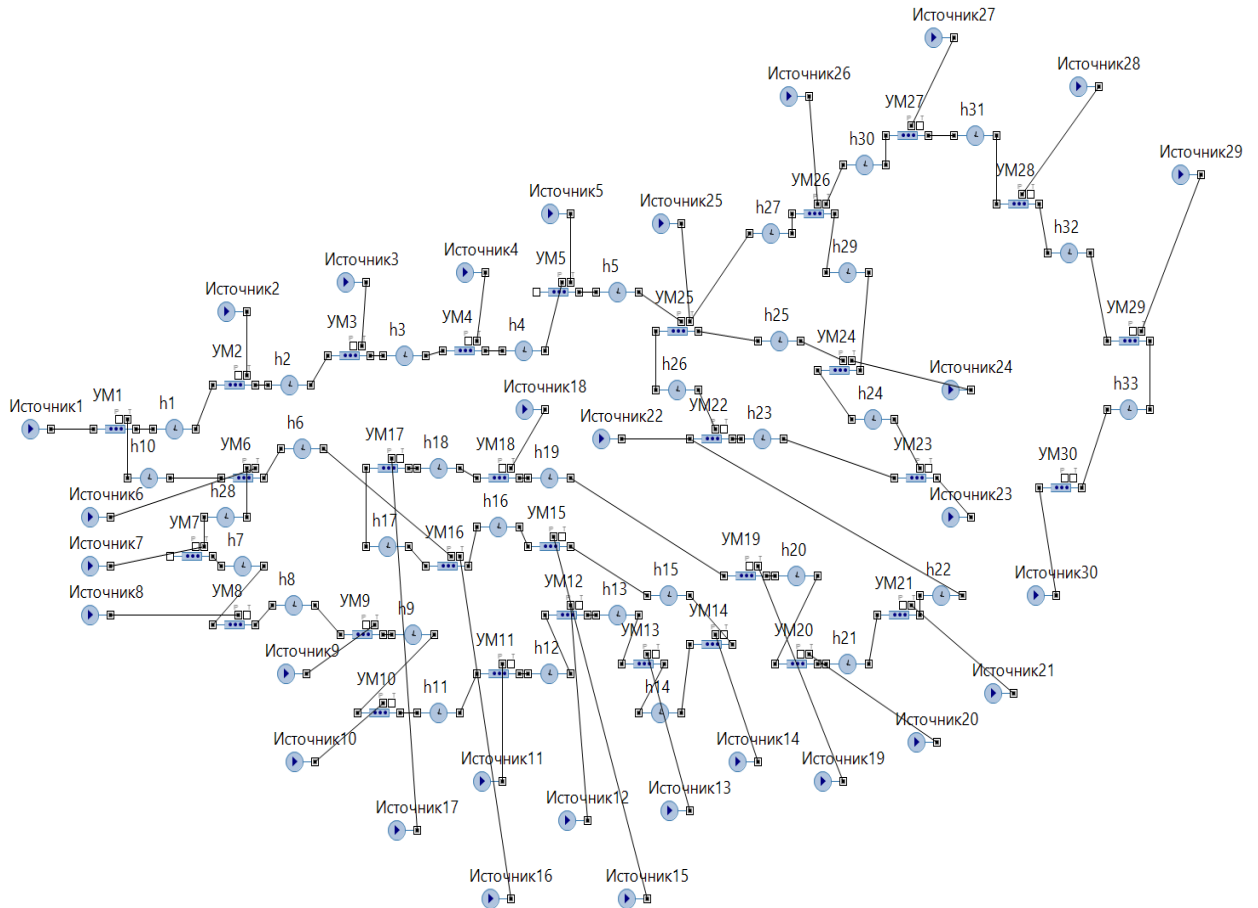


Рисунок 4.11 – Сравнительный график, показывающий результаты наблюдений для OM_{10}

Проведем такой же эксперимент для OM_{30} . На рисунке 4.12 показана имитационная модель для OM_{30} , построенная в AnyLogic.

Рисунок 4.12 – Имитационная модель для $ОМ_{30}$

В таблице 4.2 представлены обобщенные результаты наблюдений для $ОМ_{30}$.

Таблица 4.2 - Результаты наблюдений для $ОМ_{30}$

Количество моментов нестационарной нагрузки в произвольные моменты времени t	Сквозность S_K^j , секунд	
	СМц	СМдц
10	3,7	3,5
30	5,4	4,6
70	10	7,1
100	14,7	9,9
150	36,7	19,1
200	61	25,4

На рисунке 4.13 представлен сравнительный график, показывающий результаты наблюдений для OM_{30} .

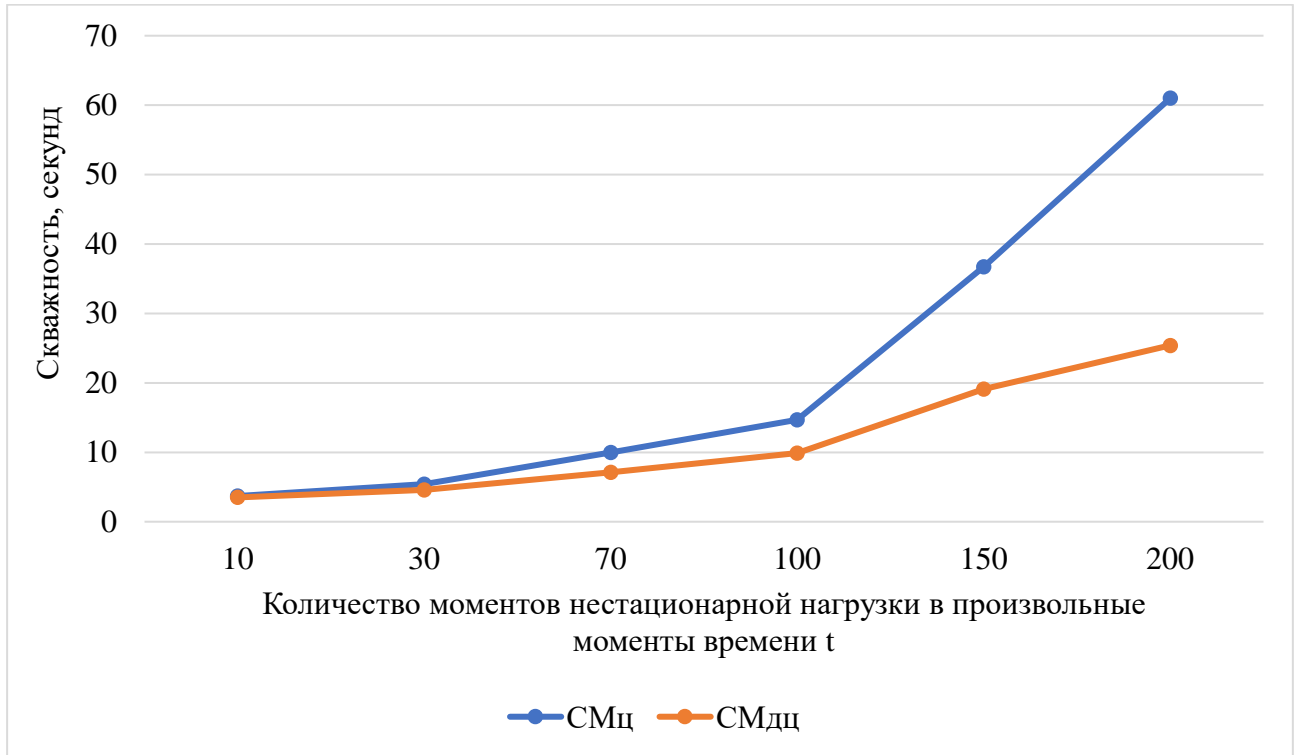


Рисунок 4.13 – Сравнительный график, показывающий результаты наблюдений для OM_{30}

Таким образом, исходя из полученных результатов наблюдения (таблица 4.1 и таблица 4.2) в сетях с различной структурой, можно сделать вывод, что разработанная децентрализованная распределенная многоагентная СМ эффективно работает в сетях с нестационарной нагрузкой.

Несмотря на положительные результаты проведенных экспериментов, разработанная распределенная многоагентная децентрализованная СМ МСС не подходит для сетей с большой связностью R . Причиной этому является множественный обмен данными и результатами их обработки с соседними ИАМ, что влияет на увеличение времени обработки $t_{обр}^j$ вновь полученных данных, так как, объем данных и частота с которой они поступают на ИАМ возрастает и, как

следствие, увеличивается скважность S_k^j . Для этого был проведен отдельный эксперимент, в котором мы постепенно увеличивали связность в OM_{30} . В таблице 4.3 приведены результаты наблюдений, а на рисунке 4.14 изображен сравнительный график, показывающий влияние связности R ОМ на скважность S_k^j .

Таблица 4.3 - Результаты наблюдений по мере увеличения связности R сети

Связность R , $1:N$	Скважность S_k^j , секунд
$N = 1$	3,6
$N = 2$	4,3
$N = 5$	8,7
$N = 10$	17,9
$N = 15$	28,1
$N = 20$	36,3
$N = 25$	44
$N = 30$	54,9

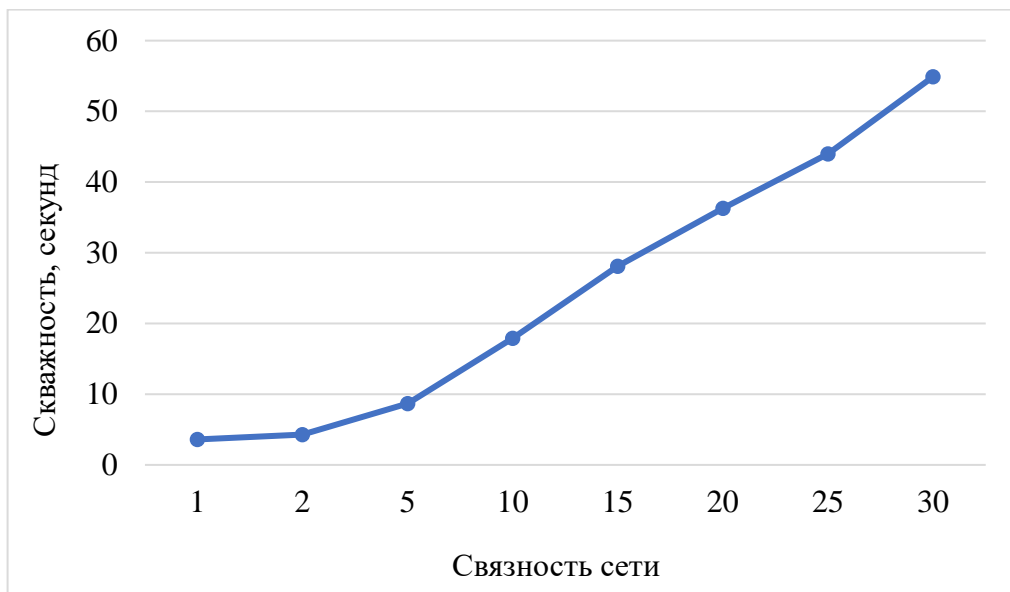


Рисунок 4.14 – Сравнительный график, показывающий влияние связности R сети на скважность S_k^j

4.4 Выводы по главе

1. Разработана двухуровневая архитектура распределенной многоагентной системы мониторинга объектов мониторинга мультисервисной сети связи в основе которой лежат два логических уровня - уровень LSTM-сети, который обеспечивает решение задачи получения такой конфигурации LSTM-сети, которая максимизирует точность прогнозирования параметров мониторинга в последующие моменты времени $t + p$ и уровень наложенного управления сообщениями Pastry, обеспечивающий формирование оверлейной сети ИАМ для решения задачи доставки сообщений с данными параметров мониторинга УМ и их прогнозных значений.

2. Разработано программное обеспечение интеллектуальных агентов мониторинга. Выбрана комбинированная аппаратная вычислительная платформа, состоящая из одноплатной вычислительной платформы «Raspberry Pi 3 Model B+» и вычислительного модуля «Intel NCS2». Приведена логическая структура ИАМ и подробно описана их техническая реализация.

3. Проведен численный эксперимент в сравнении с существующими централизованными СМ, показаны преимущества и недостатки разработанного решения.

ЗАКЛЮЧЕНИЕ

Диссертационная работа посвящена разработке средств специального математического и программного обеспечения для систем мониторинга различных предметных областей, функционирующих в условиях нестационарной нагрузки, имеющей временной лаг распространения между узлами мониторинга. Для повышения эффективности процесса мониторинга в этих условиях в работе была предложена архитектура распределенной системы мониторинга, основанной на принципах децентрализованного многоагентного управления.

Научная задача, решенная в диссертации, может базироваться на ряде научных методов из предметных областей многоагентных систем, теории марковских процессов, теории нейронных сетей и других, примененных для предметной области систем мониторинга.

Достоверность и обоснованность полученных результатов подтверждается научно организованными экспериментами, корректным применением известных методов исследования, адекватных природе изучаемых процессов и явлений, непротиворечивостью и воспроизводимостью результатов, полученных в процессе сравнительного анализа вычислений и натурных экспериментов.

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Проведён анализ состояния проблемы мониторинга объектов различного класса, функционирующих в условиях нестационарной нагрузки, которая имеет временной лаг распространения, с целью выявления общих принципов процесса их мониторинга.

2. Разработана нейросетевая модель интеллектуального агента системы мониторинга, которая обеспечивает формирование матрицы связности гиперпараметров нейронной сети и накопленных значений параметров состояния объектов мониторинга с учетом характеристик временного ряда распространения нестационарной нагрузки

3. Разработан обобщенный алгоритм динамической настройки параметров нейронной сети в заданной точке мониторинга, который минимизирует временные затраты на обучение при заданном значении точности прогнозирования.

4. Разработан обобщенный алгоритм взаимного информационного согласования интеллектуальных агентов в распределенной многоагентной системе мониторинга, который обеспечивает сокращение времени задержки передачи данных о моментах возникновения нестационарной нагрузки соседним интеллектуальным агентам, расположение TM соответствует направлению распространения временного лага нагрузки.

5. В качестве показательного примера полученных решений разработана архитектура распределенной многоагентной системы мониторинга объектов мультисервисной сети связи, которая за счет двухуровневой схемы организации, поддерживает логическую целостность данных в процессе взаимного информационного согласования подмножества входящих в нее интеллектуальных агентов.

6. Проведены сравнительные численные эксперименты по оцениванию временной задержки управляющего на узлы мониторинга для централизованной и разработанной архитектур системы мониторинга.

Полученные результаты могут найти применение в проектировании и разработке действующих вариантов систем мониторинга для различных предметных областей, обеспечивающей функции мониторинга в условиях возникновения нестационарной нагрузки в узлах мониторинга. Направления дальнейших исследований могут заключаться в совершенствовании методов и алгоритмов взаимодействия множества интеллектуальных агентов, обеспечивающих балансировку информационных и вычислительных ресурсов системы мониторинга для более широкого спектра условий нестационарности нагрузки на подконтрольных узлах мониторинга.

СПИСОК ТЕРМИНОВ, СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

IoT – интернет вещей

ITS – intelligent transportation systems – интеллектуальные транспортные системы

ТДК – транспортно-дорожный комплекс

МСС – мультисервисная система связи

ЦОД – центр обработки данных

СХД – система хранения данных

ОУ – объект управления

СМ – система мониторинга

ОМ – объект мониторинга

УМ – узел мониторинга

SLA – service level agreement – уровень предоставления услуг

KPI – key performance indicator – ключевой показатель эффективности

IPTD – IP packet transfer delay – задержка передачи IP-пакетов

IPDV – IP packet delay variation – изменение задержки IP-пакета

IPLR – IP packet loss ratio – коэффициент потери IP-пакетов

INMS – integrated network management system – интегрированная система управления сетью связи

DCIM – data center infrastructure management – система управления центром обработки данных

ЛО – ливневые осадки

СВ – система водосброса

TAU – tracking period updating – обновление периода отслеживания

UE – user equipment – пользовательское оборудование

PSM – power saving mode – режим энергосбережения

M2M – machine-to-machine – межмашинное взаимодействие

DPP – day peak period – дневной пиковый период

FDMI – fixed day measuring interval – фиксированный интервал измерений за день

СММ – скрытые марковские модели

DBN – deep belief network – глубокая сеть доверия

RNN – Recurrent neural network – рекуррентная нейронная сеть

LSTM – long short-term memory – долгая краткосрочная память

UUID – universally unique identifier – всемирно уникальный идентификатор

МАС – многоагентная система

ИА – интеллектуальный агент

БДО – база данных онтологии

OWL – ontology web language – веб-язык онтологии

ВИС – взаимное информационное согласование

ИАМ – интеллектуальный агент мониторинга

ОРО – обратное распространение ошибки

МППР – марковский процесс принятия решений

БД – база данных

СУБД – система управления базами данных

СМО – система массового обслуживания

ВОА – bayesian optimization approach – Байесовский подход к оптимизации

NMM – neural minimization methods – методы нейронной минимизации

ДНП – динамическая настройка параметров

ДДНП – децентрализованная динамическая настройка параметров

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Internet of things [Электронный ресурс], режим доступа: https://en.wikipedia.org/wiki/Internet_of_things (дата обращения 11.01.2022).
2. Интеллектуальные транспортные системы. Термины и определения. // ГОСТ Р 56829-2015.
3. Directive 2010/40/EU of the European Parliament and of the Council of on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. // Official Journal of the European Union. 6.8.2010. p. 207/1- 207/13.
4. Stawiarska, E., Sobczak, P. The Impact of Intelligent Transportation System Implementations on the Sustainable Growth of Passenger Transport in EU Regions. // Sustainability 2018, 10, 1318. p. 1-32.
5. Крюков О.В., Остриков А.Ю., Щербаков В.С. Модели и методы для обеспечения качества обслуживания в мультисервисных системах связи: монография. – Орел: Академия ФСО России, 2017. – 149 с.
6. Корнеев Н.В., Гончаров В.А. Анализ моделей SAAS, IAAS, PAAS CRM-систем. // Интернет-журнал "Технологии техносферной безопасности", Выпуск № 2 (60), 2015 г. 1-10 С.
7. Mell P., Grance T. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. 2011. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
8. Генеративно-состязательная сеть [Электронный ресурс], режим доступа: https://ru.wikipedia.org/wiki/Генеративно-состязательная_сеть (дата обращения 14.02.2022).
9. Основы генеративно-состязательных сетей [Электронный ресурс], режим доступа: <https://habr.com/ru/articles/726254/> (дата обращения 14.02.2022).
10. Industrial IoT (IoT) Platform Market Size, Share & Trends Report 2022-2028 [Электронный ресурс], режим доступа: <https://www.researchgate.net/publication/365823785> (дата обращения 10.04.2022).

11. Analysis of network traffic features generated by IoT devices [Электронный ресурс], режим доступа: <https://www.researchgate.net/publication/339253470> (дата обращения 10.04.2022).
12. Обзор автомобильного рынка в России в 2022 году и перспективы развития [Электронный ресурс], режим доступа: <https://data.tedo.ru/automotive/tedo-russia-automotive-market-overview-2022.pdf> (дата обращения 25.04.2022).
13. Состояние и перспективы развития рынка электротранспорта (автомобили, платформы, топливные элементы) в России и мире. Оценка влияния на показатели развития НТИ "Автонет" [Электронный ресурс], режим доступа: https://www.aggf.ru/projects/Аналитические%20отчеты%202022/Отчет%20по%20состоянию%20и%20перспективам%20развития%20рынка%20электротранспорта%20в%20России%20и%20мире_2022.pdf (дата обращения 25.04.2022).
14. Self-economy in Cloud Data Centers: Statistical Assignment and Migration of Virtual Machines [Электронный ресурс], режим доступа: <https://www.researchgate.net/publication/220768407> (дата обращения 15.05.2022).
15. Tsvetkov V. Ya. Global Monitoring // European Researcher, 2012, Vol. (33), № 11-1, p.1843- 1851.
16. Калачев Д. Н. Информационный мониторинг. – 2014. Вестник МГТУ МИРЭА – №. 2, выпуск 3. – С. 301-311.
17. Общие принципы построения вычислительных сетей [Электронный ресурс], режим доступа: https://neo-chaos.narod.ru/useful/network/olifer_nets-7.pdf (дата обращения 18.07.2022).
18. Service-level agreement [Электронный ресурс], режим доступа: https://en.wikipedia.org/wiki/Service-level_agreement (дата обращения 11.08.2022).
19. Система мониторинга wiSLA [Электронный ресурс], режим доступа: <https://www.wellink.ru/products/wisla> (дата обращения 24.08.2022).
20. Минайчев А.А. Управление качеством обслуживания в системах обработки информации на основе гистерезисного метода с двумя типами порогов. Диссертация на соискание ученой степени кандидата технических наук, 2018.

21. Руководство по Zabbix [Электронный ресурс], режим доступа: <https://www.zabbix.com/documentation/5.4/ru/manual> (дата обращения 2.09.2022).
22. M. Ancona, N. Corradi, A. Dellacasa, G. Delzanno, other. On The Design Of An Intelligent Sensor Network For Flash Flood Monitoring, Diagnosis And Management In Urban Areas // The 4th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS-2014), Procedia Computer Science 32 (2014) 941 – 946.
23. Крюков О.В. Метод параметрической идентификации трафика на ограниченном интервале наблюдения / О.В. Крюков, В.С. Щербаков // Промышленные АСУ и контроллеры. – 2019. – № 4. – С. 9–18.
24. J. Conklin, "Effect of control lag on performance in a tracking task," *Journal of Experimental Psychology*, vol. 53, pp. 261-268, 1957.
25. B. Watson, N. Walker, W. Ribarsky, and V. Spaulding, "Effects of variation in system responsiveness on user performance in virtual environments," *Human Factors*, vol. 40, no. 3, pp. 403-414, Sept.1998.
26. Narrowband IoT [Электронный ресурс], режим доступа: https://en.wikipedia.org/wiki/Narrowband_IoT (дата обращения 05.10.2022).
27. Recommendation E.500 [Электронный ресурс], режим доступа: <https://www.itu.int/rec/T-REC-E.500-198811-S/en> (дата обращения 05.10.2022).
28. Recommendation E.492 [Электронный ресурс], режим доступа: <https://www.itu.int/rec/T-REC-E.492-199602-I/en> (дата обращения 05.10.2022).
29. Штепан Е.А. М2М и автомобили станут причиной роста трафика [Электронный ресурс], режим доступа: <https://nag.ru/news/25881> (дата обращения 05.10.2022).
30. Ложковский А. Г. Модель трафика в мультисервисных сетях с коммутацией пакетов // Наукові праці ОНАЗ ім. О. С. Попова. – 2010. - № 1. – С. 63-67.
31. Алиев Т. И., Аппроксимация вероятностных распределений в моделях массового обслуживания // Научно-технический вестник информационных технологий, механики и оптики. – 2013. – №. 2 (84). – С 88-93.

32. Крюков О.В., Козачок В.И., Баранов В.А., Кривенцев Д.И. Оценивание показателей качества обслуживания транспортной сети с пакетной коммутацией на основе скрытой марковской модели // *Электросвязь*. – 2014. – № 5. – С. 48–53.
33. Okutani, I., Stephanedes, Y.J.: ‘Dynamic prediction of traffic volume through kalman filtering theory’, *Transport. Res. B Methodol.*, 1984, 18, (1), pp. 1–11.
34. Крюков, О.В. Метод параметрической идентификации трафика на ограниченном интервале наблюдения / О.В. Крюков, В.С. Щербаков // *Промышленные АСУ и контроллеры*. – 2019. – № 4. – С. 9–18.
35. Hidden Markov Models: Estimation and Control / Elliott, R.J.; Aggoun, L.; Moore, J.B. 1995, XIV, 382 p. Hardcover, ISBN 0-387-94364-1.
36. Терентьев В.М., Паращук И.Б. Теоретические основы управления сетями многоканальной радиосвязи. – СПб.: Типография ВАС, 1995. – 195 с.
37. Баранов, В. А. Модель мультимедийного трафика, использующая многомерные плотности распределения вероятности наблюдаемых состояний / В. А. Баранов, О. В. Крюков, В. С. Щербаков // *Информация и космос*. – 2015. – № 2. – С. 28-35.
38. Г. И. Ивченко, Ю. И. Медведев, Смеси вероятностных распределений и случайные размещения, *Тр. по дискр. матем.*, 1998, том 2, 169–182.
39. Н. А. Волков, С. А. Буденный, А. М. Андрианова. Смеси вероятностных распределений в задачах регрессии и проверки на аномальность и их применение для PVT-свойств. // *ТРУДЫ МФТИ*. 2020. Том 12, № 3, С. 17-43.
40. Турилова Е.А., Халиуллин С.Г. Математические и вероятностные основы финансовых расчетов. Часть II. Стохастический анализ финансового рынка/ Е.А. Турилова, С.Г. Халиуллин. – Казань: Казан. ун-т, 2015. – 86 с.
41. M. Daily, S. Medasani, R. Behringer, and M. Trivedi, “Self-driving cars,” *Computer*, vol. 50, no. 12, pp. 18–23, 2017.
42. Huang, W., Song, G., Hong, H., et al.: ‘Deep architecture for traffic flow prediction: deep belief networks with multitask learning’, *IEEE Trans. Intell. Transport. Syst.*, 2014, 15, (5), pp. 2191–2201.

43. Lv, Y., Duan, Y., Kang, W., et al.: 'Traffic flow prediction with big data: A deep learning approach', *IEEE Trans. Intell. Transp. Syst.*, 2015, 16, (2), pp. 865–873.
44. Ma, X., Yu, H., Wang, Y., et al.: 'Large-scale transportation network congestion evolution prediction using deep learning theory', *PLoS One*, 2015, 10, (3), pp. 1–17.
45. Lingras, P., Sharma, S., Zhong, M.: 'Prediction of recreational travel using genetically designed regression and time-delay neural network models', *Transport. Res. Record*, 2002, 13, (1), pp. 435–446.
46. Hochreiter, S., Schmidhuber, J.: 'Long short-term memory', *Neural Comput.*, 1997, 9, (8), pp. 1735–1780.
47. Ma, X., Tao, Z., Wang, Y., et al.: 'Long short-term memory neural network for traffic speed prediction using remote microwave sensor data', *Transport. Res. C Emerging Technol.*, 2015, 54, pp. 187–197.
48. A Simple Network Management Protocol (SNMP). Network Working Group, Request for Comments: 1157 [Электронный ресурс], режим доступа: <https://datatracker.ietf.org/doc/html/rfc1157> (дата обращения 11.07.2022).
49. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Network Working Group, Request for Comments: 3411 [Электронный ресурс], режим доступа: <https://datatracker.ietf.org/doc/html/rfc3411> (дата обращения 11.07.2022).
50. Введение в JSON. ECMA-404 The JSON Data Interchange Standard. [Электронный ресурс], режим доступа: <http://json.org/json-ru.html> (дата обращения 07.09.2022).
51. Structure of Management Information Version 2 (SMIPv2). Network Working Group, Request for Comments: 3411 [Электронный ресурс], режим доступа: <https://datatracker.ietf.org/doc/html/rfc2578> (дата обращения 11.07.2022).
52. Gerhard Weiss, ed. by, *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.
53. Michael Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons Ltd, 2002, paperback, 366 pages.

54. Лихтенштейн В. Е., Конявский В. А., Росс Г. В., Лось В. П. Мультиагентные системы. Самоорганизация и развитие. / «Финансы и статистика». – М., 2018. – 264 с.
55. OWL Web Ontology Language Semantics and Abstract Syntax: 3411 [Электронный ресурс], режим доступа: <https://www.w3.org/TR/2004/REC-owl-semantic-20040210/> (дата обращения 11.07.2022).
56. Емельянов В. В., Ясиновский С. И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО / Емельянов В. В., Ясиновский С. И. - М.: АНВИК, 1998. - 426 с.
57. Coulouris G., Dollimore J., Kindberg N. Distributed Systems: Concepts and Design (3rd Edition). Addison-Wesley, 2001, с. 452.
58. С.С. Еськов, Е.В. Лебеденко, О.Я. Кравец. Управление взаимным информационным согласованием в системах распределенного реестра на основе специального математического и программного обеспечения: Монография/ – Воронеж: Издательство «Научная книга», 2021. – 120 с.
59. Рожкова Т.С. Специальное математическое и программное обеспечение децентрализованного управления динамической реконфигурацией вычислительной системы на базе мобильных устройств. Диссертация на соискание ученой степени кандидата технических наук, 2022.
60. Сигорский В.П. Математический аппарат инженера. Изд. 2-е, стереотип. «Техніка», 1977, 768 с.
61. Гибсон А., Паттерсон Д. Глубокое обучение с точки зрения практика. Издательство «ДМК-Пресс», 2018. – 418 с.
62. Fernandez S., Graves A., Schmidhuber J. Sequence labelling in structured domains with hierarchical recurrent neural networks // Proc. 20th Int. Joint Conf. on Artificial Intelligence, Ijcai 2007: journal. — 2007. — P. 774—779.
63. Pascanu R., Mikolov T., Bengio Y. On the difficulty of training recurrent neural networks. // Proceedings of the 30 th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP, volume 28.

64. Рыкшин М.С. Разработка модели прогнозирования ключевых показателей эффективности в распределенной системе информационного мониторинга с децентрализованной структурой / М.С. Рыкшин, А.М. Бойченко, О.Ю. Миронов, О.Я. Кравец // Системы управления и информационные технологии, №1(87), 2022. – С. 36-40.
65. J. Elman, Finding structure in time. *Cognitive Science*, 14(2) (1990) 179-211.
66. Рыкшин М.С. Математическая модель распределенной автоматизированной системы информационного мониторинга в режиме реального времени в условиях нестационарной нагрузки / М.С. Рыкшин // Системы управления и информационные технологии. – 2021. – № 4 (86). – С. 69-72.
67. Hochreiter S., Bengio Y., Frasconi P., Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
68. Агаммедова С.А., Дышин О.А., Асланов Дж.Н., Габибов И.А. Марковские процессы принятия решений по управлению рисками в технических системах / *The scientific heritage*, № 79 (2021), С. 31-36.
69. Bertsekas, D. Value and Policy Iteration in Optimal Control and Adaptive Dynamic Programming / *IEEE Transactions on Neural Networks*, Report LIDS-P-3174, May 2015, pp. 1-10.
70. Rowstron A., Kermarrec A., Castro M., Druschel P. Scribe: The design of a large-scale event notification infrastructure. *Networked Group Communication*, Third International COST264 Workshop (NGC'2001), November 2001.
71. Рыкшин М.С. Разработка математического обеспечения распределённой системы мониторинга при нестационарной нагрузке на объекте мониторинга / М.С. Рыкшин, Т.С. Рожкова, О.А. Морозова // Системы управления и информационные технологии. – 2023. - №2(92). – С. 66-70.
72. Rykshin M.S., Morkovin S.V., Filimonov A.V., Tsvetkova O.L. Comparative Analysis of Metrics for Estimating the Introduced Distortions in Images when Injection

Digital Watermarks in the Frequency Spectrum // AIP Conference Proceedings. Manuscript № AIPCP21-AR-MIP 2021-00155.

73. Рыкшин М.С., Лебеденко Е.В., Волков С.А., Минайчев А.А. Программа для выявления скрытых закономерностей в больших данных на основе платформы распределенных вычислений – ApacheSpark. - Свидетельство о государственной регистрации программы для ЭВМ № 2019667174. - М: ФИПС, 2019.

74. Рыкшин М.С. Подходы к разработке алгоритма взаимного информационного согласования интеллектуальных агентов в распределенной многоагентной системе мониторинга / М.С. Рыкшин // Моделирование, оптимизация и информационные технологии. – 2023. – № 11 (3). Режим доступа: <https://moitvvt.ru/ru/journal/article?id=1367>.

75. Lee M., Lin J., Gran E. DistTune: Distributed Fine-Grained Adaptive Traffic Speed Prediction for Growing Transportation Networks. Transportation Research Record, May 2021.

76. Shaikh A., Jamal M., Hanif F., Khan M., Inayatullah S. Neural minimization methods (NMM) for solving variable order fractional delay differential equations (FDDEs) with simulated annealing (SA). October, 2019.

77. Рыкшин М.С., Воробьев А.В., Куцакин М.А., Белоусов В.Н. Исследование методов многофакторной аутентификации в распределенных автоматизированных системах мониторинга и аналитической обработки информации / М.С. Рыкшин, А.В. Воробьев, М.А. Куцакин // Информационные технологии моделирования и управления. – 2019. – Т. 116. – № 2. – С. 134-141.

78. Rykshin M.S., Rojkova T.S., Karpenko O.A. Research of distributed storage systems and methods of distributed processing for big data // Modern informatization problems in simulation and social technologies (MIP-2020'AS). Proceedings of the XXV-th International Open Science Conference, Yelm, WA, USA. – January 2020. – p. 235-239.

79. Keras API Reference [Электронный ресурс], режим доступа: <https://keras.io/api/> (дата обращения 11.12.2022).

80. Statistic models, hypothesis tests and data exploration [Электронный ресурс], режим доступа: <https://www.statsmodels.org/stable/index.html> (дата обращения 25.12.2022).

81. Руководство по TensorFlow [Электронный ресурс], режим доступа: <https://www.tensorflow.org/guide?hl=ru> (дата обращения 3.01.2023).

82. Cisco Packet Tracer [Электронный ресурс], режим доступа: http://elib.bsut.by/bitstream/handle/123456789/932/cisco_packet_tracer_network_simulator_-_2016.pdf?sequence=1&isAllowed=y (дата обращения 9.01.2023).

83. Учебные пособия по AnyLogic [Электронный ресурс], режим доступа: <https://anylogic.help/ru/tutorials/index.html> (дата обращения 17.01.2023).

84. Rykshin M.S. To the question of the relevance development of the traffic monitoring system for IOT infrastructure on the basis of packet switched network// Modern informatization problems in the technological and telecommunications systems analysis and synthesis (MIP-2021'AS). Proceedings of the XXVI-th International Open Science Conference, Yelm, WA, USA. – January 2021. – p. 153-157.

85. Крюков О.В., Ульянов И.В. Модель логического канала мультисервисной сети связи, использующей ресурс оператора связи в условиях недостаточных наблюдаемости и управляемости. Информационные системы и технологии. – 2019. – № 3(113). – С. 97–104.

86. Крюков О.В., Щербаков В.С., Разепов И.И. Повышение качества обслуживания в мультисервисной сети связи, использующей ресурс оператора связи. Применение искусственного интеллекта в информационно-телекоммуникационных сетях. Сборник материалов научно-практической конференции. – 2021. – С. 134–146.