

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

На правах рукописи



ГОРШКОВ Алексей Владиславович

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ПРОЦЕССОВ УПРАВЛЕНИЯ ИНФОКОММУНИКАЦИОННЫМИ
СЛУЖБАМИ НА ОСНОВЕ СИСТЕМ МОБИЛЬНЫХ АГЕНТОВ**

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

Диссертация

на соискание учёной степени кандидата технических наук

Научный руководитель:
д.т.н., профессор Кравец Олег Яковлевич

Воронеж – 2023

Оглавление

Введение	3
Основное содержание работы	9
1. Проблемы управления распределенными информационными системами на основе мультиагентных технологий.....	14
1.1. Задачи распространения информации в мультиагентной среде	14
1.2. Управление нагрузкой в распределенных системах и парадигма мобильных агентов	20
1.3. Миграция услуг на основе мобильных агентов для обеспечения устойчивости к внешним и внутренним возмущениям	27
1.4. Построение деревьев агрегации данных для увеличения срока службы сети в многоагентных системах	31
1.5. Архитектура составных мобильных приложений	37
1.6. Постановка задач работы	40
2. Исследование механизма распространения информации в мультиагентной системе на основе марковских процессов	42
2.1. Ожидаемое время распространения	42
2.2. Распространение во временном окне	55
2.3. Управление процессом	61
2.4. Выводы	64
Список литературы к главе 2	67
3. Мобильные программные агенты как инструмент управления нагрузкой в распределенных инфокоммуникационных системах ...	69
3.1. Мобильные агенты и балансировка нагрузки	69
3.2. Результаты эксперимента	76
3.3. Миграция услуг на основе мобильных агентов для обеспечения устойчивости к внешним и внутренним возмущениям	86
3.4. Выводы	99
Список источников к главе 3	103
4. Проектирование архитектур сложных многоагентных систем	107
4.1. Увеличение срока службы сети в многоагентных системах на основе построения деревьев агрегации данных	107
4.2. Высокоуровневое описание архитектуры составных мобильных приложений на основе метамоделирования	127
4.3. Проектирование механизма межагентного взаимодействия	138
4.4. Выводы	142
Список источников к главе 4	145
Заключение	151
Список использованных источников.....	152

Введение

Актуальность **темы.** Стремительное развитие инфокоммуникационных служб (объектов развитых сложноструктурированных информационных систем) породило множество методов управления ими. Большой вклад в развитие методов управления внесли Крупин А., Черняк Л., Kale P.A., Liu X., Wang J. В теоретическом плане ряд методов управления инфокоммуникационными службами сводится к рационализации распределения вычислителей для решения внешних задач. Альтернативой здесь выступает концепция мобильных агентов с расширением алгоритмов балансировки нагрузки для улучшения равномерности ее распределения.

Несмотря на развитую теорию управления распределенными службами инфокоммуникационных сред, существуют проблемы, связанные с учетом работоспособности исполнительных платформ и динамической реконфигурацией служб. В данной области ведущие исследователи Вишневский В.М., Жожикашвили В.А., Babiker S.F., Shroff N.V. развили множество подходов, ориентированных на обычные задачи реконфигурации структуры платформ со статическим распределением приоритетов пользователей. Ключевые показатели для этих подходов, к сожалению, не учитывают то, что задачи планирования платформ решаются в реальном масштабе времени, и статической информации о состоянии платформ оказывается недостаточно. В результате оказывается необходимой архитектура системы мобильных агентов с динамическим учетом корректности работы платформ и динамическим перемещением служб для повышения доступности услуг

Мобильные инфокоммуникационные системы используют значительную долю служебного трафика на получение полной информации о коммуникационной подсистеме. Однако такая полнота далеко не всегда нужна, ситуация зависит от задач и требований к

параметрам систем. Следовательно, необходимо разработать алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных на основе алгоритмов восстановления пути для снижения накладных расходов на взаимодействие и требований к полноте информации о сети на приемнике.

Таким образом, актуальность темы диссертационного исследования продиктована необходимостью дальнейшего развития средств специального математического и программного обеспечения процессов управления инфокоммуникационными службами (объектами ИС) на основе реализации систем мобильных агентов.

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

Целью работы является разработка средств специального математического и программного обеспечения процессов управления инфокоммуникационными службами (объектами ИС) на основе реализации систем мобильных агентов.

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

- построить марковскую модель распространения информации мобильными агентами в звездообразной сети, учитывающую информированности агентов и позволяющую описать одновременную передачи информации между агентами;
- разработать алгоритмы балансировки нагрузки для улучшения равномерности ее распределения, основанные на использовании мобильных агентов;
- предложить архитектуру системы мобильных агентов с динамическим учетом корректности работы платформ и динамическим

перемещением служб для повышения доступности услуг;

- разработать алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных на основе алгоритмов восстановления пути для снижения накладных расходов на взаимодействие и требований к полноте информации о сети на приемнике;

- предложить архитектуру составных мобильных приложений с использованием высокоуровневого описания системы мобильных агентов и реализацией процесса гетерогенной композиции мобильного приложения с учетом требований заказчика.

Объект исследования: процессы управления инфокоммуникационными службами (объектами ИС).

Предмет исследования: математического и программного обеспечения процессов управления инфокоммуникационными службами на основе систем мобильных агентов.

Методы исследования. При решении поставленных в диссертации задач использовались методы теории вероятностей, теории принятия решений, а также методы объектно-ориентированного программирования.

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

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

1. Марковская модель распространения информации мобильными

агентами в звездообразной сети, отличающаяся учетом информированности агентов и возможностью описания одновременной передачи информации между агентами и обеспечивающая уменьшение вычислительной сложности расчета ожидаемого времени распространения с $O(N^3)$ до $O(N^2)$.

2. Алгоритмы балансировки нагрузки, основанные на парадигме мобильных агентов и обеспечивающие улучшение равномерности распределения нагрузки.

3. Архитектура системы мобильных агентов, отличающаяся динамическим учетом корректности работы платформ и обеспечивающая динамическое перемещение служб для повышения доступности услуг.

4. Алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных, отличающийся применением эвристического и «жадного» алгоритмов восстановления пути и обеспечивающий снижение накладных расходов на взаимодействие и снижением требований к полноте информации о сети на приемнике.

5. Архитектура составных мобильных приложений, отличающаяся высокоуровневым описанием системы мобильных агентов и обеспечивающая реализацию процесса гетерогенной композиции мобильного приложения с учетом требований заказчика.

Теоретическая и практическая значимость исследования заключается в развитии средств специального математического и программного обеспечения математического и программного обеспечения процессов управления инфокоммуникационными службами на основе на основе систем мобильных агентов, а также информационного и программного обеспечения для экспериментальной оценки качества разработанных методов и алгоритмов.

Теоретические результаты работы могут быть использованы в

проектных и научно-исследовательских организациях, занимающихся проектированием платформенно-инвариантных систем управления инфокоммуникационными службами в условиях штатной или нерегламентированной внешней нагрузки.

Положения, выносимые на защиту

1. Марковская модель распространения информации мобильными агентами в звездообразной сети обеспечивает уменьшение вычислительной сложности расчета ожидаемого времени распространения с $O(N^3)$ до $O(N^2)$.

2. Алгоритмы балансировки нагрузки улучшают равномерность распределения нагрузки.

3. Архитектура системы мобильных агентов обеспечивает динамическое перемещение служб для повышения доступности услуг.

4. Алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных снижает накладные расходы на взаимодействие и требования к полноте информации о сети на приемнике.

5. Архитектура составных мобильных приложений реализует процесс гетерогенной композиции мобильного приложения с учетом требований заказчика.

Результаты внедрения. Основные результаты внедрены в Научно-исследовательском институте вычислительных комплексов им. М. А. Карцева» (г. Москва) при проектировании инфокоммуникационных систем на основе мобильных приложений, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Апробация работы. Основные положения диссертационной работы

докладывались и обсуждались на следующих конференциях: 9-й Всероссийской научно-технической конференции с Международным участием «Прогрессивные технологии и процессы» (Курск, 2022); 11-й Всероссийской научно-практической конференции «Решение» (Пермь, 2022); XXVIII-th International Open Science Conference Modern informatization problems in simulation and social technologies (Yelm, WA, USA, 2023), II Всероссийской научной конференции «Достижения науки и технологий-ДНиТ-11-2023» (Красноярск, 2023); XXVIII-th International Open Science Conference «Modern informatization problems in the technological and telecommunication systems analysis and synthesis» (Yelm, WA, USA, 2023), а также на научных семинарах кафедры автоматизированных и вычислительных систем ВГТУ (2021-2023 гг.).

Достоверность результатов обусловлена корректным использованием теоретических методов исследования и подтверждена результатами сравнительного анализа данных вычислительных и натурных экспериментов.

Публикации. По результатам диссертационного исследования опубликовано 13 научных работ, в том числе 6 – в изданиях, рекомендованных ВАК РФ (из них 2 – в изданиях WoS и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [2] - марковская модель распространения информации мобильными агентами в звездообразной сети; [1, 7, 10] - алгоритмы балансировки нагрузки, основанные на парадигме мобильных агентов; [6, 8, 9] - архитектура системы мобильных агентов, отличающаяся динамическим учетом корректности работы платформ; [4] - алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных; [3, 5] - информационное и программное обеспечение

для экспериментальной оценки качества разработанных методов и алгоритмов.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 206 наименований. Работа изложена на 172 страницах.

Основное содержание работы

Во введении обоснована актуальность исследования, сформулированы его цель и задачи, научная новизна и практическая значимость полученных результатов, приведены сведения об апробации и внедрении работы.

В первой главе исследуются особенности разработки математического и программного обеспечения управления инфокоммуникационными службами на основе систем мобильных агентов. Отмечено, что повысить эффективность управления инфокоммуникационными службами можно путем модификации моделей распространения информации мобильными агентами в звездообразной сети, алгоритмов балансировки нагрузки для улучшения равномерности ее распределения, создания архитектуры системы мобильных агентов с динамическим учетом корректности работы платформ и динамическим перемещением служб, разработки алгоритма расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных на основе алгоритмов восстановления пути. Потребовалась формализация данных задач, а также алгоритмизации их решения. Сформулирована цель и задачи исследования.

Вторая глава посвящена формализации механизмов распространения информации в мультиагентной системе на основе аппарата марковских процессов.

В работе исследуется процесс распространения информации в

мультиагентной системе, в котором каждый агент представлен цепью Маркова с двумя состояниями - L и M. Состояние L относится к «дому», тогда как состояние M относится к «месту встречи». Когда два агента остаются вместе в M, они «встречаются» и образуют контакт. Это означает, что они могут обмениваться информацией, осуществлять иные взаимодействия.

В общем случае процесс можно классифицировать как процесс распространения с мобильными агентами, и он позволяет продемонстрировать несколько интересных свойств модели. Предоставлен эффективный способ вычисления времени распространения и исследована зависимость процесса распространения от таких параметров, как количество агентов, количество неинформированных агентов в конце процесса и интенсивность контакта.

В любой момент времени либо все узлы, либо никто из тех, кто находится в M, не проинформированы. Состояние системы может быть описано тремя переменными: число информированных агентов в L, обозначаемое i ; количество неосведомленных агентов в L - обозначается j ; и состояние осведомленных, неосведомленных $N - i - j$ агентов, которые находятся в M.

Также рассмотрены процессы перемещения информации, которая актуальна только в пределах данного временного окна $[0, t]$, и после этого интервала агенты перестают распространять информацию. Исследовано количество агентов, получивших информацию в течение временного окна, и вероятность того, что процесс завершится до t после того, как все агенты будут проинформированы.

Показано, что ожидаемое время, необходимое для информирования дополнительного агента, сначала мало, а затем увеличивается, а вероятность информирования всех агентов в течение заданного срока имеет S-образную форму. Также предоставлена информация о том, как на

процесс может повлиять изменение параметров моделирования, таких как начальное и конечное количество информированных агентов и интенсивность контактов.

Третья глава посвящена исследованию и разработке мобильных программных агентов как инструментов управления нагрузкой в распределенных информационных системах.

Предложен вклад в балансировку нагрузки на основе парадигмы мобильного агента. Предполагается, что узлы находятся в «кооперативной» сети, что позволяет изучать только характеристики предлагаемого алгоритма, и предполагается, что задачи независимы, т.е. между ними нет связи.

Предложены три типа агентов: стационарного агента и двух мобильных агентов:

- Агент-наблюдатель (ОА).
- Мобильный агент-супервайзер (SMA).
- Мобильный агент-транспортёр (ТМА).

Детально рассмотрены алгоритмы их работы. Синтез алгоритмов дает результирующий алгоритм балансировки нагрузки. Показана его эффективность.

Далее рассмотрена проблема миграции услуг на основе мобильных агентов. В системах, которые оказывают услуги в реальном масштабе времени, где показатель QoS крайне важен для пользователя, возникают ситуации перегрузки платформ, на которых функционирует информационное и программное обеспечение, оказывающее услуги. Поддержание QoS на требуемом уровне требует перемещения услуг между платформами, от перегруженных к недогруженным. Для эффективного решения задачи перемещения необходим набор методов и инструментов. Целью является создание методологии перемещения в реальном масштабе

времени с поддержанием QoS на требуемом уровне.

Коллективная работа агентов порождает план перемещения, учитывая параметры надежности, безопасности и загруженности платформ. Важным условием является то, что при перемещении служб не произойдет никаких изменений у остальных агентов. Фактически такое перемещение агентов сохраняет не только параметры QoS, но и обеспечивает повышение отказоустойчивости всей системы.

В главе 4 представлены особенности проектирования архитектур сложных многоагентных систем.

Увеличение времени жизни сети (NL) является важным требованием в беспроводных сенсорных сетях (WSN). Одним из методов увеличения NL является использование деревьев агрегации данных (DAT). DAT улучшают NL, комбинируя преимущества энергоэффективности как агрегации данных (DA), так и маршрутизации на основе дерева. Пока для построения DAT широко используются как централизованные, так и распределенные стратегии, мы предлагаем комбинированный подход для улучшения NL. Этот подход снижает накладные расходы на связь и ослабляет потребность в полной информации о сети на приемнике. В этой работе такой комбинированный подход называется подходом расширенного локального просмотра (ELV). На основе ELV предложены два алгоритма построения DAT: ELV с фиксированным приемником (ELVF) и ELV со случайным приемником (ELVR). Оба алгоритма используют основанный на эвристике метод восстановления локального пути (LPR) и жадный алгоритм расширенного восстановления пути (EPR). Используя эти методы планируется последовательность DAT, которые в совокупности улучшают NL, а также уменьшают накладные расходы на реструктуризацию DAT. Эффективность ELVF и ELVR оценивается с помощью строгих экспериментов, и результаты моделирования

показывают, что предложенные алгоритмы улучшают NL и масштабируются при различных значениях коэффициента DA.

Далее в главе рассмотрена архитектура составных мобильных приложений. Несмотря на большое количество доступных мобильных приложений, потребности пользователей в повседневной жизни отличаются друг от друга. Эти проблемы создают некоторую необходимость в механизме композиции мобильных приложений для достижения желаемых функциональных возможностей с учетом различной контекстной информации (т.е. характеристик программного и аппаратного обеспечения мобильного устройства). Используя механизм композиции, можно решить ключевую задачу разработки программного обеспечения: возможность повторного использования существующих объектов.

Представлен композиционный процесс для создания контекстно-зависимых гетерогенных мобильных приложений с учетом стоимости композиции. Основная цель: во-первых, создавать мобильные приложения с использованием существующих гетерогенных программных объектов путем обеспечения процесса гетерогенной композиции, чтобы максимизировать преимущества существующих функциональных возможностей и, таким образом, удовлетворить требования пользователей. Во-вторых, учет контекстной информации мобильного устройства во время композиции, чтобы разрабатывать мобильные приложения, которые могут воспринимать и адаптироваться к среде их исполнения.

Разработан прототип подсистемы проектирования механизма межагентного взаимодействия. Программная реализация прошла государственную регистрацию в ФИПС.

В заключении представлены основные результаты работы.

1. ПРОБЛЕМЫ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ ИНФОРМАЦИОННЫМИ СИСТЕМАМИ НА ОСНОВЕ МУЛЬТИАГЕНТНЫХ ТЕХНОЛОГИЙ

1.1. Задачи распространения информации в мультиагентной среде

Сущность в процессе распространения, такая как например информация, передается между агентами по механизмам связи между ними. Литература о процессах распространения обширна и с ней можно ознакомиться с помощью [2.9, 2.13, 2.14, 2.19].

В работе исследуется процесс распространения информации в мультиагентной системе, в котором каждый агент представлен цепью Маркова с двумя состояниями - L и M. Состояние L относится к «дому», тогда как состояние M относится к «месту встречи». Когда два агента остаются вместе в M, они «встречаются» и образуют контакт. Это означает, что они могут обмениваться информацией, осуществлять иные взаимодействия. Предположим, что изначально все находятся в состоянии L, и ровно один из агентов обладает информацией. В общем случае процесс можно классифицировать как процесс распространения с мобильными агентами, и он позволяет продемонстрировать несколько интересных свойств модели. Предоставлен эффективный способ вычисления времени распространения и исследована зависимость процесса распространения от таких параметров, как количество агентов, количество неинформированных агентов в конце процесса и интенсивность контакта.

Общепринятый подход к моделированию процессов распространения подразумевает сеть, в которой отдельные элементы представлены узлами, а соседние узлы взаимодействуют друг с другом и обмениваются информацией (будем использовать данную терминологию далее). В любой момент времени узел, который содержит информацию,

может играть роль распространителя, распространяя информацию среди своих соседей. Другие узлы могут представлять неосведомленных агентов (восприимчиво-инфекционная модель) и информированных агентов, которые не способны к дальнейшей передаче информации (восприимчиво-инфекционная восстановленная модель). Существует множество вариаций модели, включающих в себя случайности, изменения статуса, как представлено в [2.3]. Центральные исследуемые вопросы включают в себя время, затраченное на передачу информации от узла к другим узлам [2.5] или характеристику состояния системы.

В частности, рассматриваем конечное множество из N мобильных узлов (агентов), каждый из которых перемещается между своими индивидуальными локациями $L_1, \dots, L_N$. Общее место встречи обозначается как M . Движение в сети контролируется при помощи непрерывной сети Маркова. Время, затраченное в состоянии L_i , экспоненциально распределено с параметром λ , после чего агент переходит в M . Продолжительность времени в M экспоненциально распределена с параметром μ , после чего агент возвращается в исходное состояние L_i . Изначально все N узлов находятся у своих истоков; одному из них сообщается информация. Информированный агент в месте встречи M делится информацией с другими агентами, находящимися в M . Интересует способ распространения информации и, в частности, время, пока все узлы не будут проинформированы, то есть время распространения.

Базовый граф сети представляет собой звездообразный граф с единственным основным хаб-узлом M , представляющим место встречи, и листьями $L_1, \dots, L_N$, представляющими локации агентов. Упростим модель, объединив источники $L_1, \dots, L_N$ в одну локацию L , в которой не создаются взаимодействия и не передается информация. Таким образом, вместо того, чтобы рассматривать звездный граф, имеем просто сеть с двумя узлами M

и L.

Похожую модель рассматривают в [2.6], в которой роль лидера играет один из узлов популяции. Авторы предлагают формулу для ожидаемого времени до встречи лидера с любым заданным числом узлов не лидеров.

В моделях эпидемий часто используется безразмерное репродуктивное число $R=\lambda(k)/\mu$, где $\lambda(k)$ - скорость, с которой человек с k контактами распространяет болезнь, а μ - индивидуальная скорость выздоровления. Аналогичным образом используем безразмерную меру контактной интенсивности $\rho=\lambda/\mu$, дающую отношение скорости индивидуального перехода от L к M к скорости обратного движения. Большая интенсивность контактов означает, что время пребывания в M значительно больше времени, проведенного в L, а интенсивность взаимных контактов высока.

Общепринятые модели процессов распространения существенно отличаются от рассматриваемых. Эти процессы либо достигают устойчивого состояния, либо заканчиваются в одном поглощающем состоянии (например, когда все люди излечиваются). В нашей модели есть одно поглощающее состояние, когда все узлы проинформированы.

Представляемая модель принадлежит к семейству процессов распространения с подвижностью. Обзор соответствующей литературы представлен в разделе IX работы [2.14]. В большинстве этих моделей предполагается, что в каждом узле сети находится один агент, тогда как в [2.2] исследуются системы метапопуляций, где в каждом узле сети размещается несколько агентов, которые перемещаются по дугам сети. Другие статьи также предполагают марковские движения. Например, предлагаемую модель можно рассматривать как частный случай общей модели [2.16]. Там объединяется мобильность сети и распространение болезни следующим образом: у каждого человека есть место

происхождения (узел в сети). Индивиды перемещаются в сети в соответствии с матрицей перехода (миграции), которая имеет специальный компонент для возвращения в исходный узел. Процесс заражения также носит общий характер и зависит как от количества контактов с другими в узле и соседних с ним узлах, так и от самого узла. В нашей модели люди всегда возвращаются из M в места своего происхождения L_i , и мы предполагаем, что заражение гарантировано в состоянии M и невозможно в L_i .

Большая часть предшествующей литературы посвящена имитации реальных процессов распространения. Это приводит к сложным сетям и типам взаимодействий, которые невозможно аналитически решить даже для простых частных случаев. Наша же цель состоит в том, чтобы предложить новый подход и проиллюстрировать, как он работает, исследуя упрощенный случай. Благодаря этой простоте можно дать определенные ответы на основные вопросы, касающиеся поведения процесса распространения.

Возможно, наиболее близкой к нашему подходу является статья [2.7] и, в частности, протокол неограниченного множественного копирования, где исходный агент изначально обладает информацией, которую необходимо доставить агенту-получателю. Каждая пара агентов внутри группы, в которую входят эти двое агентов, встречается на очень короткое время в интервалах экспоненциальной длины и обменивается информацией. Основной вопрос заключается в характеристике распределения времени от источника до места назначения. В [2.7] выводятся формулы для аппроксимации этой модели.

Основная особенность нашей модели, которая отличается от традиционных моделей процессов распространения, заключается в том, что мы допускаем одновременное взаимодействие узлов (передачу информации, заражение), тогда как общепринятое соглашение

предполагает единичные взаимодействия. Следовательно, наша модель не определяет процесс рождения-смерти. Процесс распространения на звездном графике также рассматривается в [2.12].

Теперь представляем план исследования вместе с основными результатами и идеями.

Увеличение числа агентов N имеет два противоположных эффекта на параметр времени распространения: больше агентов должно быть проинформировано, но также больше агентов служит для распространения информации. Будет продемонстрирована интересная зависимость времени распространения от количества агентов в двух крайних случаях интенсивности контакта. Когда интенсивность контакта очень высока, время распространения пропорционально сумме гармоник $H(N)$ и увеличивается с N . Напротив, когда интенсивность контакта очень мала, время распространения пропорционально $H(N)/N$ и убывает с N . Исследуется зависимость для промежуточных контактных интенсивностей, показывая, что в зависимости от контактной интенсивности функция может быть монотонно возрастающей или сначала убывающей, а затем возрастающей, за исключением значений $N=2$, $N=3$, где она всегда возрастает.

Будет представлен эффективный способ нахождения ожидаемого времени распространения, который вытекает из интересного представления диаграммы переходов состояний. Вместо решения системы линейных уравнений $O(N^2)$ достаточно решить $N-1$ систему из N уравнений, что явно является улучшением, если для решения уравнений не существует алгоритма линейной сложности.

При заданном размере популяции N ожидаемое время, необходимое для информирования следующего агента, увеличивается с увеличением числа уже информированных агентов. Если позволить небольшой части агентов оставаться в неведении, это значительно сократит время

распространения.

Установлено, что вероятность того, что все агенты будут информированы ко времени t как функция от t , имеет S-образную форму. Более того, ожидаемое количество клиентов, проинформированных к данному моменту времени t , имеет S-образную форму. Этот вывод согласуется с теорией распространения инноваций, подробно описанной в [2.15]. Эта теория возникла из наблюдения, что в многочисленных исследованиях в различных областях скорость принятия инновации имеет S-образную форму.

С тех пор этот вывод был подтвержден во многих других исследованиях и подтвержден теоретическими моделями. Случайную выборку примеров в различных областях, демонстрирующих эмпирическое и теоретическое повсеместное существование S-образной кривой, можно увидеть в статьях [2.1, 2.4, 2.8, 2.10, 2.11, 2.17, 2.20].

Также исследуются возможные зависимости процессов распространения, такие как зависимость времени распространения от интенсивности контакта; интенсивность контакта для достижения целевого времени распространения; эффект от увеличения начального размера графа, то есть количества агентов, которые изначально являются проинформированными. Данный набор аналогичен проблеме максимизации влияния, рассмотренной в [2.18]. Мы наблюдаем, что изначальное количество проинформированных узлов оказывает сильное влияние, когда количество либо слишком мало, либо очень велико, а эффект довольно умеренный между узлами.

1.2. Управление нагрузкой в распределенных системах и парадигма мобильных агентов

Распределенным системам необходимо выполнять балансировку нагрузки на своих хостах, чтобы вычисления выполнялись как можно быстрее. Исследования в этой области показали появление парадигмы мобильных агентов в качестве перспективного решения. В настоящей работе эта парадигма используется, чтобы предложить подход к балансировке нагрузки, который использует преимущества мобильности агентов, в частности, на этапе сбора информации. Необходимо иметь общее системное видение при одновременном снижении затрат на сетевую связь, а также других преимуществ, таких как отказоустойчивость и расширяемость для крупномасштабных сетей. Таким образом, цель состоит в том, чтобы улучшить распределение нагрузок сбалансированным образом, чтобы максимально приблизить нагрузки к средней нагрузке системы. Результаты экспериментов показывают эффективность предлагаемого подхода к балансировке нагрузок и сокращению времени отклика.

Тенденция компьютерного мира к "распределенным системам" больше не предусматривает работу одного компьютера без взаимодействия или сотрудничества с другими компьютерами. Эти системы должны быть спроектированы с учетом новых требований [3.1, 3.2, 2.3]. В работе представлены две взаимосвязанные области исследований: "Мобильные агенты" и "балансировка нагрузки".

Технология мобильного программного агента является одной из известных технологий в области распределенных вычислений. Он возник как альтернатива классической парадигме "клиент/сервер", которая представляет собой наиболее широко используемый подход при создании распределенных приложений [3.4, 3.5]. Технология мобильных агентов

имеет интересные перспективы для различных областей применения, среди которых мы находим электронную коммерцию, поиск информации в Интернете и область балансировки нагрузки (LB).

Для достижения лучшей производительности в распределенных системах проблема балансировки нагрузки интенсивно изучалась исследователями [3.6, 3.7, 3.8]. Балансировка позволяет максимально использовать доступные ресурсы, и это может быть достигнуто за счет разумного распределения задач.

В работе изучается использование мобильных агентов в области балансировки нагрузки. Основываясь на этом исследовании, предлагается подход, направленный на сбалансирование нагрузки на систему с использованием мобильности агентов путем принятия трех агентов и определения задач для каждого из этих агентов. Основная идея состоит в том, чтобы локальные нагрузки были близки к средней нагрузке системы.

1.2.1. Механизмы балансировки нагрузки

В [3.9] авторы используют новую концепцию передачи виртуальных нагрузок, которая позволяет узлам прогнозировать будущие нагрузки, которые они получают в последующих итерациях. Соответственно, уведомленный узел будет учитывать эту прогнозируемую нагрузку при передаче части своей фактической нагрузки некоторым из своих соседей.

В [3.10] представлен подход к планированию ресурсов и балансировке нагрузки для эффективного предоставления облачных услуг. Они расширяют использование виртуальных машин и обеспечивают балансировку нагрузки за счет динамического выбора запроса из класса с использованием алгоритма оптимизации многомерной очереди.

В [3.11] представлена методология балансировки нагрузки для решения проблемы загрузки контейнеров при автомобильных перевозках. Предлагаются генетический алгоритм со случайным ключом (BRKGA) с

новой функцией полезности, которая учитывает статическую стабильность и баланс нагрузки.

В [3.12] предложены алгоритмы динамической репликации данных (DDRA) из трех этапов для улучшения дублирования информации в облачной системе хранения. Первые два этапа заключаются в определении адекватных узлов обслуживания для достижения баланса рабочей нагрузки на основе рабочих нагрузок узлов. На третьем этапе представлена схема динамического развертывания дублирования, предложенная для обеспечения более высокой производительности доступа и лучшей балансировки нагрузки. На первом этапе для реализации начального LB фильтруются сервисные узлы с вероятностью ниже определенного порога.

В [3.13] предложен новый вариант протокола направленной диффузионной маршрутизации в беспроводных сенсорных сетях. Этот вариант пытается улучшить существующий протокол, используя механизм балансировки нагрузки, чтобы сбалансировать энергию датчиков.

1.2.2. Мобильные агенты в балансировке нагрузки

Авторы [3.14] предлагают динамическую балансировку нагрузки в P2P-сети малого мира с использованием мобильных агентов. Во-первых, они объединяют одноранговые узлы, которые имеют одинаковый набор общих ресурсов. Затем они балансируют нагрузку запросов на узлах внутри кластера, чтобы избежать перегрузки сети. Для этой цели определены три агента: Агент хоста (HA), Агент обнаружения (DA) и Агент повторного подключения (RA). DA мигрирует между внутренними узлами для обнаружения скоплений узлов. Этот агент генерируется и контролируется HA, который проходит через все группы. RA генерируется для повторного подключения связей между перегруженными узлами и недогруженными узлами одного и того же кластера. Параметр привлекательности измеряется для каждого узла в соответствии с

некоторыми критериями, такими как степень узла, производительность обработки и ресурсы, содержащиеся в узле. DA мигрирует на узел с высоким параметром привлекательности, когда DA прибывает, он проверяет, перегружен ли узел или недогружен. Однако алгоритм выбора выбирает партнерский узел, который может отказаться принимать запросы агента DA, если он перегружен. В этом случае агент DA мигрирует на другие узлы, пока не найдет нужный узел. Таким образом, правильный партнер может быть неправильно выбран до начала процесса миграции.

В [3.15] предложены две различные модели управления нагрузкой в крупномасштабных распределенных системах. Первая модель основана на мобильных агентах при сборе состояния фактических ресурсов узла (оперативной памяти и процессора). Чтобы сбалансировать нагрузки, интегрируется вторая модель, основанная на нечеткой функции. Система агентов состоит из трех модулей в качестве модуля мониторинга агентов (АММ) для сбора доступного состояния текущих ресурсов. Модуль принятия решений агентом (ADM), используемый для принятия решения о переносе, когда узел перегружен, если измеренная вероятность превышает пороговое значение. Модуль миграции агентов (АММО) выбирает узел назначения для миграции в соответствии с некоторыми критериями, такими как высокий уровень обработки и вычислительных ресурсов. Однако в модели агента при определении статуса узла (перегружен/недогружен) авторы не упомянули, как определяется или измеряется используемый порог. Таким образом, если это постоянное или переменное значение и если оно измеряется локально или глобально в соответствии с другим статусом узла.

В [3.16] балансировка нагрузки реализуется с использованием мобильных агентов, которые мигрируют через все узлы на основе коэффициента кредитной стоимости. Этот фактор определяется на начальном этапе для принятия решения о выборе узла для миграции.

Выбор узла назначения, на который будет перенесен агент, основан на сравнении локальной нагрузки с пороговым значением. Если узел назначения откажется от приема нагрузки, агент будет направлен на следующий выбранный узел. Однако способ измерения порогового значения не определен. Кроме того, агенты продолжают мигрировать, пока не найдут подходящего партнера для каждого перегруженного узла.

В [3.17] представлен алгоритм балансировки нагрузки для параллельных виртуальных архитектур. Используется агент хоста для выполнения диагностического теста для оценки производительности вычислений и задержки каждого узла. Каждая параллельная задача назначается VPU (виртуальному процессору), представленному мобильным агентом. Эти VPU обмениваются сообщениями, содержащими данные и задачи, которые необходимо выполнить. Однако связь между VPU разных узлов приводит к увеличению трафика.

В [3.18] выделяются пять популяций агентов. Авторы используют подход оптимизации муравьиной колонии для параллельного распределения задач между рабочими агентами. Например, агент диспетчера (агент балансировки нагрузки) собирает необходимую информацию для принятия решений о планировании, а затем для принятия этого решения используется подход муравьиной колонии. В контексте этого исследования планировщик используется в качестве муравья, который выбирает для текущей работы машину с более высоким уровнем феромонов. Чтобы измерить вероятность этого феромона, каждая рабочая машина отправляет через своего агента контроллера агенту диспетчера количество задач, которые у нее есть в очереди.

В [3.19] авторы объединяют алгоритм наименьшего времени первого байта (LFB) и концепцию мобильного агента. Где роль мобильного агента контролирует состояние ресурсов каждого сервера. Однако авторы сосредоточены на достижении надежного подхода без учета пропускной

способности и задержки системы.

В [3.20] представлена схема балансировки нагрузки в гетерогенных P2P-системах. Предлагается три агента. Первый мобильный агент предназначен для сбора информации, а второй - для определения партнера-получателя для перегруженного узла. Третий - стационарный агент, который находится на каждом узле для обновления своей таблицы маршрутизации. Его роль заключается в том, чтобы информировать все сетевые узлы об адресах других узлов и распространять информацию об отказе узла в сети. Этот механизм использования сообщений для распространения информации каждый раз на все узлы приводит к увеличению накладных расходов.

Алгоритм балансировки нагрузки для гетерогенных P2P-систем предложен в [3.21]. В нем используется один тип мобильного агента с его основными компонентами, такими как сбор, анализ и местоположение. Коэффициент использования измеряется в соответствии с нагрузкой и пропускной способностью каждого узла. Однако для переноса перегрузки авторы предлагают выбрать соседний узел с наименьшим коэффициентом использования. Таким образом, этот один недогруженный узел будет выбран большинством перегруженных узлов или всеми из них. Следовательно, выбранный узел будет перегружен или откажется от приема задач, что приведет к повторному запуску процесса определения местоположения и повторной попытке с другими узлами, таким образом, больше времени тратится на определение местоположения, миграцию и выполнение.

1.2.3. Управление балансировкой нагрузки

В этом разделе рассмотрим различные материалы о подходах к балансировке нагрузки, существующих в литературе. Эта область в целом все еще актуальна в связи с упомянутыми недавними подходами. В

предлагаемом подходе необходимо избежать и улучшить некоторые существующие ограничения или слабые места, упомянутые ниже. В процессе выбора это касается выбора партнерского узла, на который будет перенесена перегрузка, т.е. выбор недогруженного узла для перегруженного узла. В [3.14] авторы предлагают переносить мобильный агент (МА) между набором узлов до тех пор, пока не будет найден недостаточно загруженный узел. Когда МА поступает на узел, он проверяет, может ли этот узел получить перегрузку удержания, если он отклонит прием, МА перейдет к другому из отсортированного списка [3.14, 3.15, 3.21] и повторит те же шаги проверки. Таким образом, этот метод приводит к потере времени в процессе миграции, особенно из-за того, что МА удерживает перегрузку при миграции, что замедляет ее миграцию. Таким образом, задержка системы увеличивается, и она может найти хорошего партнера, но не подходящего. Поэтому предлагается процесс выбора, направленный на поиск подходящего узла-партнера для каждого перегруженного узла, и перед запуском миграции нагрузки, чтобы избежать упомянутых недостатков. Другими словами, агент миграции направляется непосредственно к определенному и правильному узлу назначения. Кроме того, в процессе выбора некоторые работы, такие как [3.21], предлагают выбрать узел с наименьшей нагрузкой в качестве узла-партнера. Следовательно, большинство перегруженных узлов будут выбирать одного и того же партнера, этот партнер будет перегружен, а следующие принимающие агенты будут отклонены, и им следует мигрировать, чтобы найти другого партнера. Чтобы избежать этого, в предлагается метод выбора одного недогруженного узла для каждого перегруженного узла, не вызывая его перегрузки.

В общем случае состояние узла определяется как перегруженное или недогруженное в соответствии с пороговым значением. В некоторых работах, таких как [3.15, 3.16], нет упоминания о том, как определяется

или измеряется порог. Поэтому в предлагаемом подходе измеряется пороговое значение в соответствии с глобальным состоянием, основанном на состояниях всех узлов. Это сделано для того, чтобы иметь почти одинаковые состояния для всех узлов, что является основным принципом механизма балансировки нагрузки.

В отличие от [3.19], в предлагаемом подходе необходимо не только достичь сбалансированное состояние системы, но избежать увеличения пропускной способности и уменьшить задержку. Таким образом, используются мобильные агенты и их миграция таким образом, чтобы избежать обмена сообщениями между узлами, в отличие от [3.17, 3.20].

1.3. Миграция услуг на основе мобильных агентов для обеспечения устойчивости к внешним и внутренним возмущениям

1.3.1. Задача миграции услуг

Учитывая растущую сложность и динамичность сервис-ориентированных систем, обеспечение доступности услуг, несмотря на вредоносные атаки или системные сбои, является сложной задачей. В приложениях, где доступность услуг имеет решающее значение, стратегическая миграция служб, зависящих от времени, с их текущих платформ на другие платформы является жизнеспособным решением, позволяющим обеспечить непрерывное предоставление этих услуг на новых платформах. Миграция служб требует специальной инфраструктуры и механизмов для перемещения сервисных программ и данных с одной платформы на другую. Разработка эффективной схемы миграции с необходимой системной поддержкой важна для обеспечения гарантированной миграции услуг.

Были проведены интенсивные исследования по миграции процессов,

задач и услуг для нескольких целей, включая высокую производительность [3.22, 3.23], отказоустойчивость [3.24, 3.25], адаптивность системы и гибкость ресурсов [3.26, 3.27], безопасность и живучесть [3.28, 3.29]. Однако миграция услуг на основе мобильных агентов является перспективной областью исследований. Агенты - это автономные и социальные программные объекты, используемые для разработки сложных приложений, предоставляющих услуги пользователям [3.30]. Идея, лежащая в основе концепции “агента”, заключается в активном и автономном модуле, который может сотрудничать и/или конкурировать с другими модулями/агентами и окружающей средой [3.31].

В случае вредоносной атаки и сбоя системы мобильные агенты могут перейти со своих поврежденных платформ на другие исправные платформы, чтобы предлагаемые ими услуги могли постоянно предоставляться на новых платформах, обеспечивая тем самым миграцию услуг. Миграция служб с помощью стратегического перемещения агентов помогает системе пережить повреждение хоста и повышает доступность служб. В работе описывается основанный на мобильных агентах подход к миграции служб, при котором группа агентов совместно формирует план миграции для перемещения со своих текущих платформ на другие более безопасные и надежные платформы.

Определим системную архитектуру для поддержки миграции агентов и опишем модель совместного принятия решений для группы агентов в процессе миграции.

В работе представлена схема миграции услуг на основе мобильных агентов. В случае, если некоторые платформы в системе были повреждены в результате инцидента безопасности или некоторые компоненты этих платформ функционально вышли из строя, мобильные агенты могут динамически перемещаться со своих текущих платформ на другие исправные, более надежные платформы и продолжать свою работу на этих

новых платформах, таким образом пережив инцидент безопасности или сбой системы. План совместной миграции гарантирует, что миграция агента не нарушит никаких операционных ограничений для группы агентов. Поскольку агенты являются социальными субъектами, некоторые агенты могут совместно работать с другими над одной и той же задачей. Один агент может функционально зависеть от других агентов. Предлагаемый подход эффективно проверяет, что план совместной миграции не будет нарушать ни одно из ограничений. Чтобы продемонстрировать осуществимость предлагаемой схемы миграции агентов, разработана система мобильных агентов, подтверждающая концепцию, основанную на существующей платформе мобильных агентов. Хотя моделирование проводилось на основе небольшой группы агентов, оно показывает потенциал применения предлагаемой схемы к более сложным приложениям для повышения доступности услуг.

1.3.2. Исследование миграции мобильных агентов

Агентные технологии все чаще используются в различных приложениях, и мобильные агенты, в частности, могут поддерживать адаптивность системы и доступность услуг. В этом разделе кратко изложены связанные с этим работы по миграции мобильных агентов.

В [3.32] представлена система мобильных агентов для интеграции веб-сервисов во всепроникающие сети с использованием беспроводных мобильных устройств. Предлагаемый подход отделяет физическое поведение мобильных агентов (например, миграцию и клонирование) от логики интеграции, что делает более гибким добавление или изменение физического поведения мобильных агентов с учетом условий окружающей среды. В [3.22] представлен подход для поддержки миграции агентов в гетерогенной среде. Агенты, написанные на разных языках, могут мигрировать между различными платформами. Фактический код агента не

перемещается; скорее, передается схема функциональности агента. На целевой платформе фабрика агентов генерирует новый код на основе схемы.

В [3.34] предложен метод сокращения трафика данных в результате миграции агентов. Работа сосредоточена на среде выполнения агента, которая кэширует код агента и статус агента, чтобы код и статус можно было использовать повторно при возвращении мобильного агента.

Предлагаемый подход позволяет гибко внедрять мобильные агенты. В [3.35] описан подход к оценке надежности мобильных агентских систем для MANET.

В их предложении мобильные агенты развертываются для сбора и распространения служебной информации в сети. Надежность системы мобильных агентов зависит от динамики MANET и моделей передвижения пользователей. В [3.36] разработана и внедрена мультиагентная система с использованием мобильных агентов. Процесс переговоров определяется для переназначения необходимых служб доступным узлам сети в соответствии с их состояниями. Транспортная онтология предназначена для простой обработки условий и сообщений для переговоров между статическими агентами, принимающими решения, и мобильными интеллектуальными агентами. В [3.37] описан метод тестирования веб-сервисов на основе мобильных агентов. Существующий инструмент тестирования (HP LoadRunner) интегрирован с Aglets для создания практической среды для тестирования веб-служб. В исследовании также сообщается об автоматическом подходе к генерации тестовых примеров, основанном на вычитании граничных значений и вводе ошибочных данных.

Настоящее исследование преследует цель, отличную от вышеуказанных исследовательских работ. В нем показано применение миграции мобильных агентов для обеспечения доступности услуг. Таким

образом, работа сосредоточена на том, как мобильность агентов помогает динамически перемещать службы с текущих скомпрометированных платформ на другие корректные платформы, чтобы повысить доступность услуг. Она дополняет современные исследования в области использования мобильных агентов для повышения отказоустойчивости системы.

1.4. Построение деревьев агрегации данных для увеличения срока службы сети в многоагентных системах

1.4.1. Беспроводные сенсорные сети и многоагентные системы

Увеличение времени жизни сети (NL) является важным требованием в беспроводных сенсорных сетях (WSN). Одним из методов увеличения NL является использование деревьев агрегации данных (DAT). DAT улучшают NL, комбинируя преимущества энергоэффективности как агрегации данных (DA), так и маршрутизации на основе дерева. Пока для построения DAT широко используются как централизованные, так и распределенные стратегии, мы предлагаем комбинированный подход для улучшения NL. Этот подход снижает накладные расходы на связь и ослабляет потребность в полной информации о сети на приемнике. В этой работе такой комбинированный подход называется подходом расширенного локального просмотра (ELV). На основе ELV предложены два алгоритма построения DAT: ELV с фиксированным приемником (ELVF) и ELV со случайным приемником (ELVR). Оба алгоритма используют основанный на эвристике метод восстановления локального пути (LPR) и жадный алгоритм расширенного восстановления пути (EPR). Используя эти методы планируется последовательность DAT, которые в совокупности улучшают NL, а также уменьшают накладные расходы на реструктуризацию DAT. Эффективность ELVF и ELVR оценивается с

помощью строгих экспериментов, и результаты моделирования показывают, что предложенные алгоритмы улучшают NL и масштабируются при различных значениях коэффициента DA.

Беспроводные сенсорные сети (WSN) – это информационные и коммуникационные технологии для интернета вещей (IoT), соединяющие все в умном мире. В WSN узлы-датчики собирают информацию, устанавливают путь к приемнику и отправляют данные по этому пути. В зависимости от требований приложения и возможностей узлов-датчиков, узлы напрямую отправляют данные в приемник или обмениваются данными через другие узлы, пока данные не достигнут приемника. Поскольку связь между узлами потребляет больше энергии, чем вычисления и измерения [4.1], протоколы маршрутизации, которые уменьшают количество передач по сети, подходят для использования в условиях ограниченного энергопотребления. Сокращение числа передач осуществляется применением агрегации данных (DA) – известного метода повышения энергоэффективности [4.2, 4.3]. Маршрутизация на основе дерева агрегации данных (DAT) в WSN объединяет преимущества как DA, так и маршрутизации на основе дерева и обеспечивает энергоэффективную стратегию маршрутизации, которая продлевает время жизни сети (NL). Время жизни сети NL – это время, в течение которого все узлы в сети могут передавать данные в приемник. DAT подходят для сценариев со строгими энергетическими условиями и агрессивными и критическими средами, где вмешательство человека невозможно. Экономическая эффективность DAT определяется структурой DAT и коэффициентом DA (α). Структура DAT определяет пути связи узлов для передачи данных в приемник, а коэффициент DA определяет, сколько блоков данных объединяется в узле [4.4, 4.6]. Исследователи тщательно изучают DAT, однако поиск оптимального дерева, которое удовлетворяет требованиям приложения, является сложной задачей. В этой работе исследуется

проблема идентификации и построения DAT для увеличения NL. В исследовании также анализируются значения NL, полученные путем изменения коэффициента DA α . Эта проблема называется «Дерево агрегации данных и коэффициент для увеличения времени жизни сети» (DATRL).

Подход, используемый для построения DAT, влияет на структуру DAT. На рис. 1.1 показаны три подхода к построению DAT. В первом подходе (рис. 1.1а) приемник имеет полную информацию о сети, и пути маршрутизации разрабатываются самим приемником с использованием централизованных глобальных методов. Во втором подходе (рис. 1.1б) каждый узел взаимодействует с узлами в своем диапазоне передачи и определяет свои пути, используя локальные методы. В третьем подходе (рис. 1.1в) приемник имеет частичную информацию о сети. Пути маршрутизации составляются с использованием комбинации централизованных глобальных методов в приемнике и локальных методов в отдельных узлах. Этот подход можно применять на практике, поскольку он снижает накладные расходы на связь и уменьшает потребность в полной сетевой информации на приемнике.

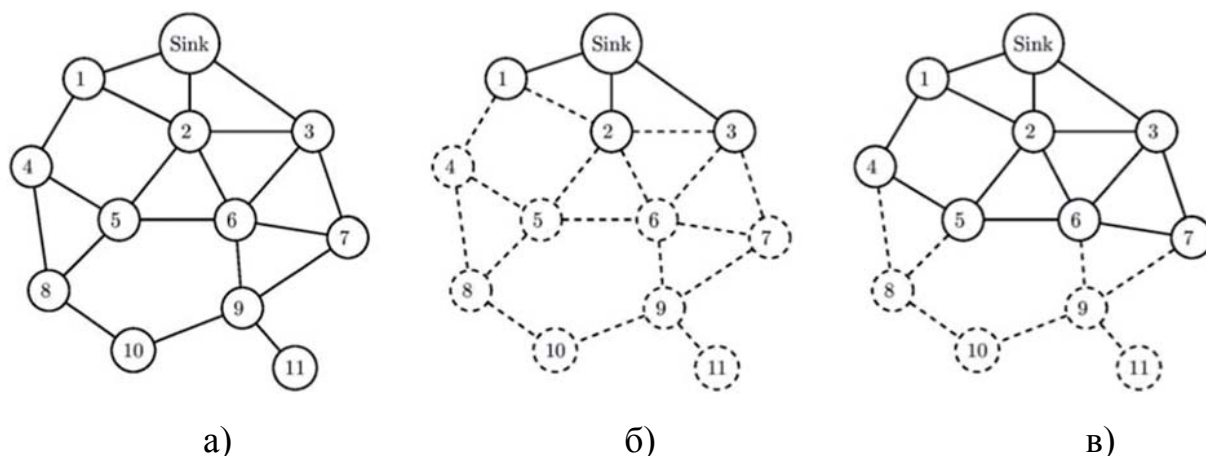


Рис. 1.1. Подходы к построению DAT

В работе совместный подход называется подходом расширенного локального просмотра (ELV), и он обосновывает предложенные методы

разработки стратегий построения DAT на его основе.

1.4.2. Предшествующие исследования

Максимизации NL способствует эффективное управление энергопотреблением с использованием подходов маршрутизации на основе дерева агрегации данных [4.2]. В этом случае NL увеличивается за счет создания DAT, в которых все узлы равномерно и оптимально потребляют энергию. Это NP-полная задача [4.7], и несколько исследователей разработали методы построения DAT, которые неоптимально распределяют энергопотребление узлов [4.6-4.17]. В [4.6] представлен алгоритм на основе реструктуризации локального дерева для максимизации NL, когда в пакет разрешено объединяться фиксированному количеству данных. В [4.7] показано, что максимизация NL является NP-полной задачей, и разрабатывают алгоритм аппроксимации для построения дерева агрегации данных. В [4.8] предложены аппроксимационные и эвристические алгоритмы для NP-полной задачи об энергоэффективной маршрутизации. В [4.9] предложена схема маршрутизации на основе минимального связующего дерева с учетом энергопотребления для улучшения NL. В [4.10] изучается проблема построения деревьев агрегации с кратчайшими путями с максимальным временем жизни и показывают, что при ограничении деревьев кратчайшими путями эта проблема может быть решена за полиномиальное время. В [4.11] представлен аппроксимационный алгоритм для построения DAT со сбалансированной нагрузкой в рамках вероятностной сетевой модели. В [4.12] обсуждается эффективная эвристика, учитывающая энергию, для генерации DAT и увеличения NL. В [4.13] обсуждается энергоэффективная агрегация данных, основанная на остовных деревьях. В [4.14] представлена эффективная эвристика на основе кластеризации для сбора данных в сенсорных сетях. Работа [4.15] использует линейное программирование и

предлагает быстрый аппроксимационный алгоритм для максимизации NL. В [4.16] обсуждается верхняя граница оптимального дерева агрегации данных и разработан алгоритм, который работает путем итеративной перестройки текущего дерева для улучшения NL. В [4.17] описана иерархическая маршрутизация в сенсорной сети с использованием энергетической метрики маршрутизации для улучшения NL. В [4.18] предлагается алгоритм аппроксимации для построения древовидной сети и уменьшения и балансировки полезной нагрузки каждого узла, что продлевает NL.

Максимизация NL, основанная на методе многоцелевой оптимизации, который одновременно максимизирует NL и минимизирует сквозную задержку, представлена в [4.19]. В [4.20] задача максимизации NL формулируется с использованием математической модели целочисленного линейного программирования и разрабатывается метод, основанный на генетическом алгоритме, для определения максимальной продолжительности жизни сети. В [4.21] предложен энергосберегающий алгоритм маршрутизации на основе нечеткой логики, который одновременно учитывает эффективность использования и балансировку энергии для улучшения NL. На основе муравьиного алгоритма в [4.22] достигается максимизация NL с минимизацией истощения энергии. Методы восстановления путей для реструктуризации DAT описаны в [4.23]. DAT, использующие алгоритм кукушки, учитывающий энергию, обсуждаются в [4.24]. В работе [4.25] обсуждается максимизация NL с использованием узлов агрегации, а в [4.26] представлено увеличение времени жизни в кластерных сенсорных сетях. В работе [4.27] разработаны эвристические методы построения DAT, увеличивающие время жизни сети, которые подходят как для идеальных, так и для реальных сценариев, с использованием индикаторов качества связи. В [4.28] обсуждаются методы построения DAT для увеличения NL с использованием

детерминированной и вероятностной модели сети. В работе [4.29] используется оставшаяся энергия узлов для повышения производительности сети. В работе [4.30] для минимизации энергопотребления предлагается настройка узлов ретрансляции и рабочего цикла с использованием схемы маршрутизации с агрегацией пакетов.

В последнее время видно, что мобильные приемники, развернутые в области датчиков, способствуют улучшению NL [4.31-4.35]. В [4.31] предлагается методика корректировки дальности связи узлов с использованием мобильного приемника для увеличения NL. В [4.32] предлагается метод кластеризации с использованием одного мобильного приемника и механизма таймера, который уменьшает проблему хот-спотов между датчиками и увеличивает NL. В [4.33] предложено увеличивать NL, используя алгоритм локальной оптимизации и мобильный приемник. В работе [4.34] представлен метод самоадаптивной кластеризации на основе метода распространения близости для сбалансированного потребления энергии в сети. В [4.35] представлен метод планирования траектории, основанный на частоте покрытия для нескольких мобильных приемников, для увеличения NL.

Структура DAT зависит от требований приложения и используемой функции агрегации. Например, в приложениях, основанных на функциях агрегации, таких как *sum* и *min*, все входящие блоки данных объединяются в один пакет [4.7-4.16, 4.36]. В исследованиях [4.6, 4.8, 4.17, 4.27, 4.37] все входящие блоки данных объединялись в несколько пакетов и соответствовали реальным условиям. Количество блоков данных, объединенных в один пакет, определяется коэффициентом DA α [4.4-4.6]. Чем больше (меньше) значение α , тем больше (меньше) блоков данных объединяются в один пакет, и, следовательно, передается меньшее (большее) количество пакетов, что приводит к меньшему (большему) потреблению энергии [4.5, 4.6, 4.27, 4.36, 4.38]. В [4.28, 4.37, 4.39]

исследовано влияние коэффициента DA на качество построенных DAT.

В исследованиях [4.5, 4.7, 4.9, 4.10, 4.36, 4.40] пути маршрутизации в DAT разрабатывались приемником с использованием централизованных глобальных методов (рис. 1.1a), в то время как в других исследованиях [4.1, 4.10-4.13] (рис. 1.1б) пути маршрутизации выбирают узлы с использованием локальных методов.

В большинстве исследований, упомянутых выше, основное внимание уделялось использованию методов максимизации NL. В этом процессе число раз, когда DAT реструктурируется, и связанные с этим накладные расходы не изучаются.

Предлагаемая работа преодолевает недостатки локальных и глобальных методов с помощью совместного подхода к построению DAT. Кроме того, эта работа увеличивает NL, планируя последовательность DAT и уменьшая связанные с этим затраты на реструктуризацию, уменьшив количество DAT.

1.5. Архитектура составных мобильных приложений

Использование и разработка мобильных приложений растут в геометрической прогрессии. Мобильная ниша представляет новые проблемы для разработки программного обеспечения [4.44]. Мобильные платформы быстро меняются, включая разнообразные возможности, такие как GPS, датчики и режимы ввода. Для этого мобильные приложения должны иметь возможность специализировать свое поведение в соответствии с контекстной информацией и, следовательно, должны быть многоканальными и работать на всех платформах.

Более того, несмотря на большое количество доступных мобильных приложений, потребности пользователей в повседневной жизни

отличаются друг от друга. Эти проблемы создают некоторую необходимость в механизме композиции мобильных приложений для достижения желаемых функциональных возможностей с учетом различной контекстной информации (т.е. характеристик программного и аппаратного обеспечения мобильного устройства). Используя механизм композиции, можно решить ключевую задачу разработки программного обеспечения: возможность повторного использования существующих объектов [4.45].

Насколько известно, не существует такого механизма композиции мобильных приложений, который учитывал бы различные факторы, такие как: вид объектов программного обеспечения, характеристики мобильных устройств, текущее состояние мобильных устройств и стоимость композиции. Таким образом, используя преимущества существующих работ, в работе представлен композиционный процесс для создания контекстно-зависимых гетерогенных мобильных приложений с учетом стоимости композиции. Основная цель: во-первых, создавать мобильные приложения с использованием существующих гетерогенных программных объектов путем обеспечения процесса гетерогенной композиции, чтобы максимизировать преимущества существующих функциональных возможностей и, таким образом, удовлетворить требования пользователей. Во-вторых, учет контекстной информации мобильного устройства во время композиции, чтобы разрабатывать мобильные приложения, которые могут воспринимать и адаптироваться к среде их исполнения.

Работа представляет собой элемент процесса создания контекстно-зависимых приложений на основе гетерогенных объектов с помощью подхода метамоделирования в мобильной среде.

Опишем несколько подходов, которые сосредоточены на механизме композиции. В [4.46] представлены некоторые проблемы, связанные с составом сервисов в мобильной среде. Описан протокол распределенной структуры услуг для мобильных сред, учитывающий мобильность,

динамично меняющуюся топологию услуг и ресурсы устройств. Работа в основном сосредоточена на распределенной архитектуре, чтобы облегчить задачу создания композиции, но не исследован ни прикладной уровень, ни его возможностях адаптации. В [4.47] представлен автоматический подход к композиции веб-сервисов на основе решения проблемы неоднородностей процессов и неоднородностей данных с помощью планировщика и посредника данных. Напротив, в [4.48] описаны соответствующие представления о существующих механизмах композиции в составной сервисной метамодели. Эта метамодель определяет все взаимосвязанные функции и обеспечивает глобальное и четкое представление о составе службы. Кроме того, этот подход позволяет конкретизировать процесс автоматической композиции: способность композита динамически изменять свою архитектуру и логику композиции в соответствии с контекстом окружающей среды. В [4.49] создана программная инфраструктура под названием AppSpotter, которая делает возможным динамический и автоматизированный выбор и состав программных компонентов для создания мобильных приложений. Предложенный механизм выбора программных компонентов основан на особенностях мобильных устройств. В [4.50] дана модель для разработки контекстно-зависимых сервисов, которые могут использоваться в качестве гибкой области для автоматического создания контекстно-зависимых композиций с помощью определенного инструмента. Кроме того, инструмент использует расширение определения проблемы, которое также включает представление контекста для расширения служб.

Эти многочисленные подходы с их функциями часто специализированы и не имеют глобального видения состава мобильных приложений. Данное исследование направлено на то, чтобы четко выразить соответствующие представления об этих существующих механизмах в процессе гетерогенной композиции для мобильных

приложений с использованием метамоделирования. Таким образом, композиция подвергается следующим специфическим ограничениям на архитектурном уровне:

- Учет ограниченных ресурсов, предлагаемых мобильными устройствами, для создания адаптивных мобильных приложений.
- Повторное использование и создание любых программных объектов для получения прибыли от существующих объектов независимо от их платформы реализации.

1.6. Постановка задач работы

Таким образом, проведенный анализ определил дизайн исследования, приведенный на рис. 1.2, определил цель и задачи исследования.

Целью работы является разработка математического и программного обеспечения управления инфокоммуникационными службами на основе систем мобильных агентов.

Для достижения цели необходимо решить следующие **задачи**:

- создать марковскую модель распространения информации мобильными агентами в звездообразной сети, учитывающую информированности агентов и позволяющую описать одновременную передачи информации между агентами;
- разработать алгоритмы балансировки нагрузки для улучшения равномерности ее распределения, основанные на использовании мобильных агентов;
- предложить архитектуру системы мобильных агентов с динамическим учетом корректности работы платформ и динамическим перемещением служб для повышения доступности услуг;



Рис. 1.2. Дизайн исследования

- разработать алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных на основе алгоритмов восстановления пути для снижения накладных расходов на взаимодействие и требований к полноте информации о сети на приемнике;

- предложить архитектуру составных мобильных приложений с использованием высокоуровневого описания системы мобильных агентов и реализацией процесса гетерогенной композиции мобильного приложения с учетом требований заказчика.

2. Исследование механизма распространения информации в мультиагентной системе на основе марковских процессов

2.1. Ожидаемое время распространения

Сущность в процессе распространения, такая как например информация, передается между агентами по механизмам связи между ними. Литература о процессах распространения обширна и с ней можно ознакомиться с помощью [2.9, 2.13, 2.14, 2.19].

В работе исследуется процесс распространения информации в мультиагентной системе, в котором каждый агент представлен цепью Маркова с двумя состояниями - L и M. Состояние L относится к «дому», тогда как состояние M относится к «месту встречи». Когда два агента остаются вместе в M, они «встречаются» и образуют контакт. Это означает, что они могут обмениваться информацией, осуществлять иные взаимодействия. Предположим, что изначально все находятся в состоянии L, и ровно один из агентов обладает информацией. В общем случае процесс можно классифицировать как процесс распространения с мобильными агентами, и он позволяет продемонстрировать несколько интересных свойств модели. Предоставлен эффективный способ вычисления времени распространения и исследована зависимость процесса распространения от таких параметров, как количество агентов, количество неинформированных агентов в конце процесса и интенсивность контакта.

Напомним, что в любой момент времени либо все узлы, либо никто из тех, кто находится в M, не проинформированы. Состояние системы может быть описано тремя переменными: число информированных агентов в L, обозначаемое i ; количество неосведомленных агентов в L - обозначается j ; и состояние осведомленных, неосведомленных $N - i - j$ агентов, которые находятся в M. Рис. 2.1 иллюстрирует переходы между состояниями, когда $N=3$.

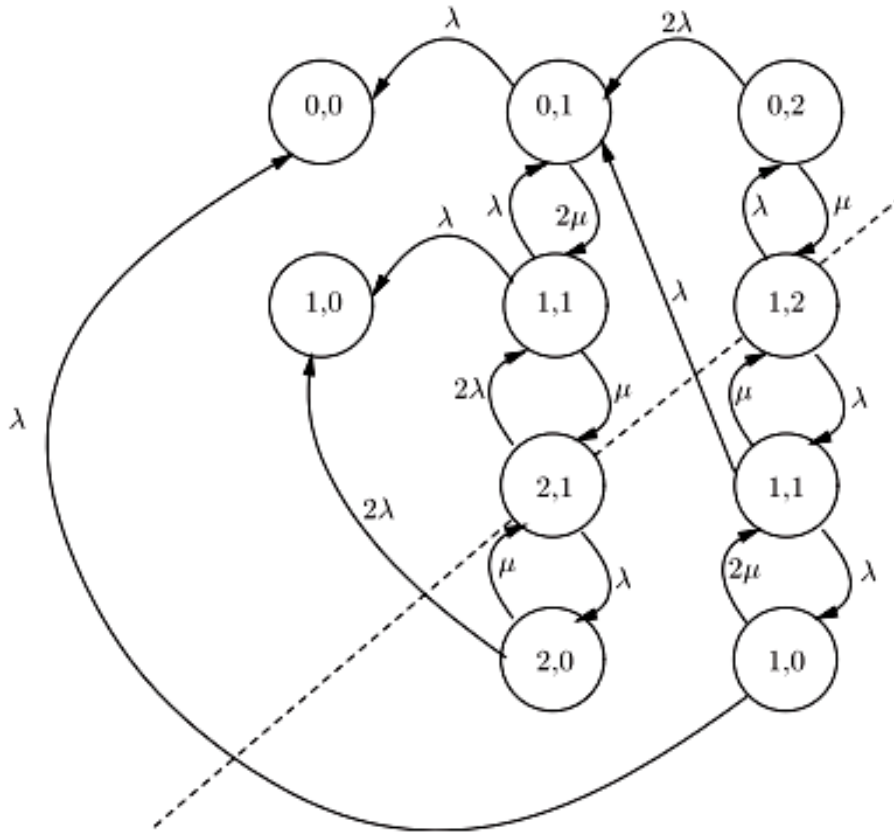


Рис. 2.1. Схема перехода для $N=3$. Состояния выше пунктирной диагонали - это состояния, в которых агенты в точке M информированы; те, что ниже диагонали - не информированы; и по диагонали агентов в M нет

Определим $T_{i,u}$ как ожидаемое время до окончания процесса распространения при наличии i информированных и u неинформированных агентов, находящихся в L . Остальные агенты (находящиеся в M) информированы. Определим $\hat{T}_{i,u}$ аналогично, но при условии, что те, кто находится в M , не информированы.

Нет нужды определять $T_{N,0}$ или $T_{N-1,0}$ т.к. в первый раз все информируются, когда процесс завершается, и в M должно быть по крайней мере 2 агента. Для $i=0, \dots, N-2$,

$$T_{i,0}=0 \quad (2.1)$$

Для $u \geq 1$ и $i+u < N$

$$[(i+u)\lambda+(N-i-u)\mu]T_{i,u}=1+i\lambda T_{i-1,u}+u\lambda T_{i,u-1}+(N-i-u)\mu T_{i+1,u} \quad (2.2)$$

(нет необходимости определять $T_{1,u}$, поскольку его коэффициент равен нулю) и

$$[(i+u)\lambda+(N-i-u)\mu]\hat{T}_{i,u}=1+i\lambda T_{i-1,u}+u\lambda \hat{T}_{i,u-1}+(N-i-u)\mu \hat{T}_{i+1,u} \quad (2.3)$$

$\hat{T}_{0,u}$ не определено, поскольку должен быть хотя бы один информированный агент.

Для $i=1, \dots, N$ и $i+u=N$,

$$N\lambda T_{i,N-i} (\equiv N\lambda \hat{T}_{i,N-i})=1+i\lambda T_{i-1,N}+(N-i-u)\lambda \hat{T}_{i,N-i-1} \quad (2.4)$$

состояние $(0,N)$ невозможно, т.к. должен быть хотя бы один информированный агент, а состояние $(N,0)$ невозможно, как объяснялось ранее.

Уравнение (2.4) является частным случаем уравнения (2.3), когда $i+u=N$. Поэтому можно было бы удалить (2.4) и изменить диапазон для (2.3) до $i+u \leq N$, отметив при этом, что $T_{i,N-i} \equiv \hat{T}_{i,N-i}$.

Основной интерес состоит в изучении свойства $T_{1,N-1} (\equiv \hat{T}_{1,N-1})$.

2.1.1. Точки экстремума

N=2

Решение, когда $N=2$ - простое и может быть использовано для иллюстрации чувствительности системы к параметрам. Уравнение перехода следующее:

$$T_{1,1} = \frac{1}{2\lambda} + \frac{1}{2} T_{0,1} + \frac{1}{2} \hat{T}_{1,0}$$

$$\hat{T}_{1,0} = \frac{1}{\lambda + \mu} + \frac{\mu}{\lambda + \mu} T_{1,1}$$

$$T_{0,1} = \frac{1}{\lambda + \mu} + \frac{\mu}{\lambda + \mu} T_{1,1}$$

Решение:

$$\begin{aligned}\frac{T_{1,1}}{1/\lambda} &= \frac{3}{2} + \frac{1}{2\rho} \\ \frac{\hat{T}_{1,0}}{1/\lambda} &= \frac{T_{0,1}}{1/\lambda} = 1 + \frac{1}{2\rho}\end{aligned}\tag{2.5}$$

где безразмерная переменная $\frac{T_{i,\mu}}{1/\lambda}$ измеряет время распространения в единицах ожидаемого времени пребывания в точке L. Таким образом, нормированное время распространения линейно возрастает в зависимости от обратной интенсивности контакта $1/\rho$.

$$\rho \rightarrow \infty$$

Рассмотрим случай $\rho \rightarrow \infty$. Это означает, что с высокой вероятностью все агенты переместятся в M раньше, чем кто-либо из них переместится обратно в L. В этом случае ожидаемое время до того, как первый агент переместится в M, равно $\frac{1}{(N-1)\lambda}$, а затем ожидаемое время $\frac{1}{(N-2)\lambda}$, до второго перемещения и так далее. Следовательно,

$$\frac{T_{1,N-1}}{1/\lambda} = \left(\frac{1}{N} + \frac{1}{N-1} + \dots + 1 \right) = H(N)$$

где H - гармоническая функция, которая асимптотически ведет себя как $\ln(N)$. Это выражение является нижней границей для $T_{i,N-i}$ для любого $\mu > 0$.

$$N \rightarrow \infty$$

Предположим, что ρ фиксировано и $N \rightarrow \infty$. В этом случае, как только информированный агент переместится в M, что занимает в среднем $1/\lambda$ времени, формируется большая группа информированных агентов из-за большого потока однородных агентов из L в M. Тогда с высокой вероятностью всегда есть хотя бы один информированный агент в M, и ситуация как в случае $\mu \rightarrow 0$; а именно, ожидаемое время до завершения

процесса равно $\frac{H(N)}{\lambda}$. Отсюда следует

$$1 < \frac{T_{L,N-1}}{1/\lambda} < H(N) + 1.$$

Нижняя граница достигается, когда μ очень мало, поэтому те, кто переезжает в M , остаются там с высокой вероятности, а верхняя граница достигается, когда μ велико, поэтому фактически подсчет начинается только после того как первоначально информированный агент перемещается в M .

$$\rho \rightarrow 0$$

Когда $\rho \rightarrow 0$, время, проведенное в M , относительно короткое. Рассмотрим типичное состояние, когда все N агенты находятся в точке L , и i из них проинформированы. Имеется $(N-i)i$ пар информированного агента и неинформированного агента и нас интересует время до встречи первой пары. При $\rho \rightarrow 0$ эти встречи могут рассматриваться как самостоятельные события. Рассмотрим заданную пару. Легко видеть, что вероятность того, что один из этих узлов находится в L , а другой в M , равна

$$2\rho/(1+2\rho+2\rho^2) \rightarrow 2\rho.$$

Среднее количество переходов в состояние, когда оба находятся в точке M , составляет $2\lambda\rho$. Учитывая, что существует $(N-i)i$ пар, и используя предположение о зависимости, скорость, с которой происходят события встречи, составляет $2\lambda\rho(N-i)i$. Ожидаемое время до первой встречи

$\frac{1}{2\lambda\rho(N-i)i}$, следовательно, и тогда у нас есть $i+1$ информированных агентов. Следовательно

$$\begin{aligned}
\frac{T_{1,N-1}}{1/\lambda} &= \sum_{i=1}^{N-1} \frac{1}{2\rho(N-i)i} = \\
&= \frac{1}{2\rho N} \sum_{i=1}^{N-1} \left(\frac{1}{(N-i)} + \frac{1}{i} \right) = \frac{H(N-1)}{\rho N}
\end{aligned} \tag{6}$$

2.1.2. Аналитические построения

На рис. 2.2 показана диаграмма переходов процесса распространения. Над диагональю расположены состояния, в которых агенты в M информированы, в состояниях ниже диагонали неинформированны, а на диагонали имеем состояния без агентов в точке M . Индексация выше диагонали отличается от индексации ниже диагонали: координаты i и u меняются местами и получают дополнительные значения, которые в сумме дают N . Длинные пунктирные стрелки указывают переходы из состояний, в которых те, M неинформированы, а информированный агент перемещается из L в M , одновременно информируя всех уже находящихся там агентов.

Замечание 2.1. Столбец j диаграммы переходов, начиная с отсчета $j=0$, соответствует множеству состояний с общим числом j неосведомленных агентов и $N-j$ информированных агентов. Чтобы увидеть это свойство, заметим, что в состояниях над диагональю в этом столбце есть j неосведомленных агентов (остальные $N-j$ агентов проинформированы, поскольку эти состояния соответствуют $T_{1,N-1}$), тогда как в состояниях под диагональю $N-j$ проинформированных агентов (остальные j агенты не информированы, т.к. эти состояния соответствуют $\hat{T}_{1,N-1}$).

Заметим, что переходов слева направо нет; следовательно, вычисление может выполняться по одному столбцу за раз слева направо. В расчетах используем общую систему индексации для всех записей. Таким образом, запись (p, q) относится к состоянию в строке p и столбце q

диаграммы переходов, как показано на рис. 2.2. Диапазон индексов равен $p=0, 1, \dots, N$ и $q=0, 1, \dots, N-1$. Таким образом, для $p+q < N$ состояние выше диагонали соответствует значению T ; если $p+q > N$, то состояние ниже диагонали соответствует значению $\hat{T}$; когда $p+q=N$, T и $\hat{T}$ совпадают.

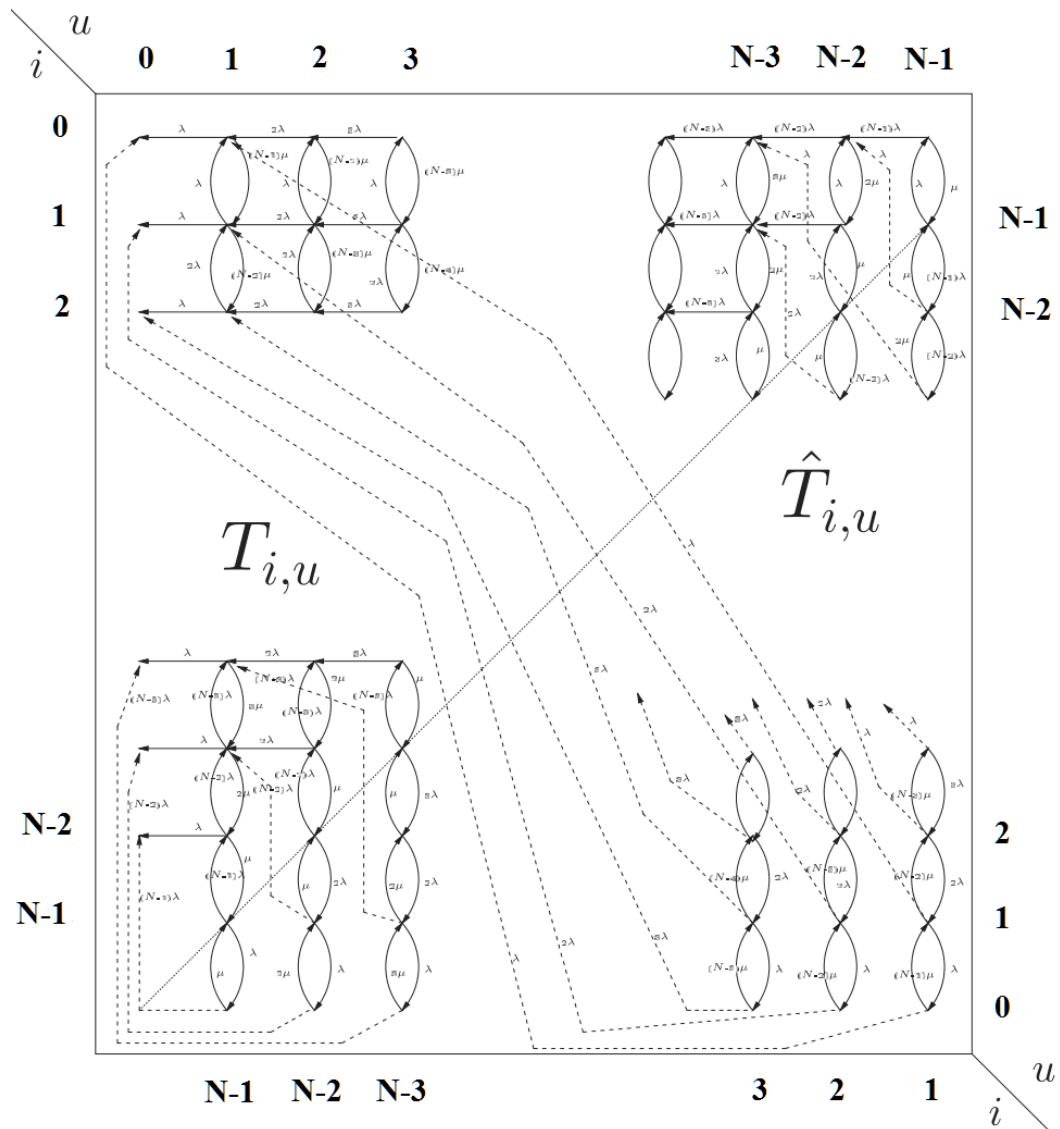


Рис. 2.2. Схема перехода. Состояния выше диагонали - это состояния, в которых агенты в точке M информированы; те, что ниже диагонали, не информированы; а по диагонали агентов в M нет

Начальные условия для крайнего левого столбца - $T_{i,0}=0$. Предположим, что значения T и $\hat{T}$ соответствующие столбцам $j < q$, уже вычислены. Теперь вычисляем значения, соответствующие столбцу q ,

следующим образом. Пусть X_p - ожидаемое время до завершения процесса, когда система находится в состоянии, соответствующем строке p в столбце q (исключаем q из этого обозначения, поскольку предполагаем, что оно фиксировано). Таким образом, при $p \leq q$ имеем $X_p = T_{p,q}$; а при $p \geq q$ имеем $X_p = \hat{T}_{N-q, N-p}$.

Теперь запишем уравнения (2.2)-(2.4) через неизвестные X_p и уже известные переменные $T_{i,u}$, соответствующие столбцам $j < q$. Преобразование уравнений (2.2) и (2.4) является прямым, и просто заменяем i на p . Для каждого $p=0, \dots, N-q-1$, выражение (2.2) принимает вид

$$-p\lambda X_{p-1} + [(p+q)\lambda + (N-p-q)\mu] - (N-p-q)\mu X_{p+1} = 1 + q\lambda T_{p,q-1} \quad (2.7)$$

Для $p=N-q$ уравнение (2.4) принимает вид

$$-p\lambda X_{p-1} + N\lambda X_p - q\lambda X_{p+1} = 1$$

Необходимо соблюдать осторожность при записи уравнения для $p=N-q+1, \dots, N$. В этом случае при работе с нижними диагональными элементами, соответствующими значениям $\hat{T}$, используем соотношения $q=N-i$ и $p=N-u$. Здесь $\hat{T}_{i,u}$ соответствует неизвестному X_p , $\hat{T}_{i,u+1}$ соответствует X_{p-1} , $T_{i-1,u}$ уже посчитанная $T_{N-q-1, N-p}$. Выражение (2.3) принимает вид:

$$-(p+q-N)\mu X_{p-1} + [(2N-q-p)\lambda + (p+q-N)\mu] X_p - (N-p)\lambda X_{p+1} = 1 + (N-q)\lambda T_{N-q-1, N-p} \quad (2.8)$$

При этом X_1 и X_{N+1} не нужно определять, потому что они появляются с нулевым коэффициентом.

Таким образом, многократно решается система из $N+1$ линейных уравнений с $N+1$ неизвестными, а затем увеличивается u на единицу, пока не получим полное решение системы.

На рис. 2.3 показано ожидаемое время распространения для различных значений N и μ . Нормализуем время в числовые примеры так, что $\lambda=1$. Нижний график соответствует $\mu=0,1$; как объяснено ранее, $T_{1,N-1}$ приблизительно соответствует гармонической последовательности.

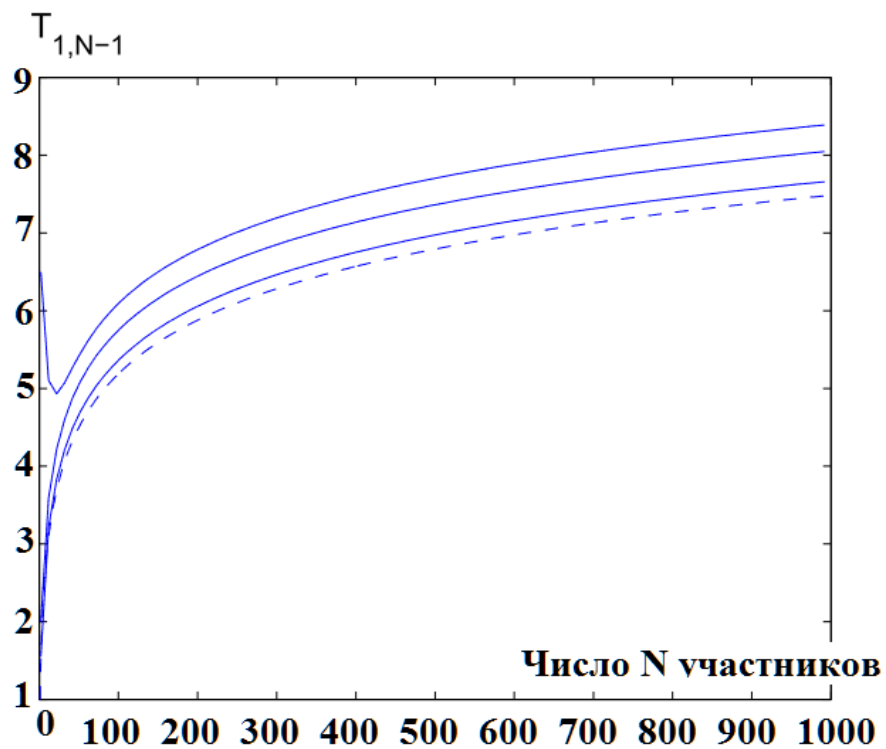
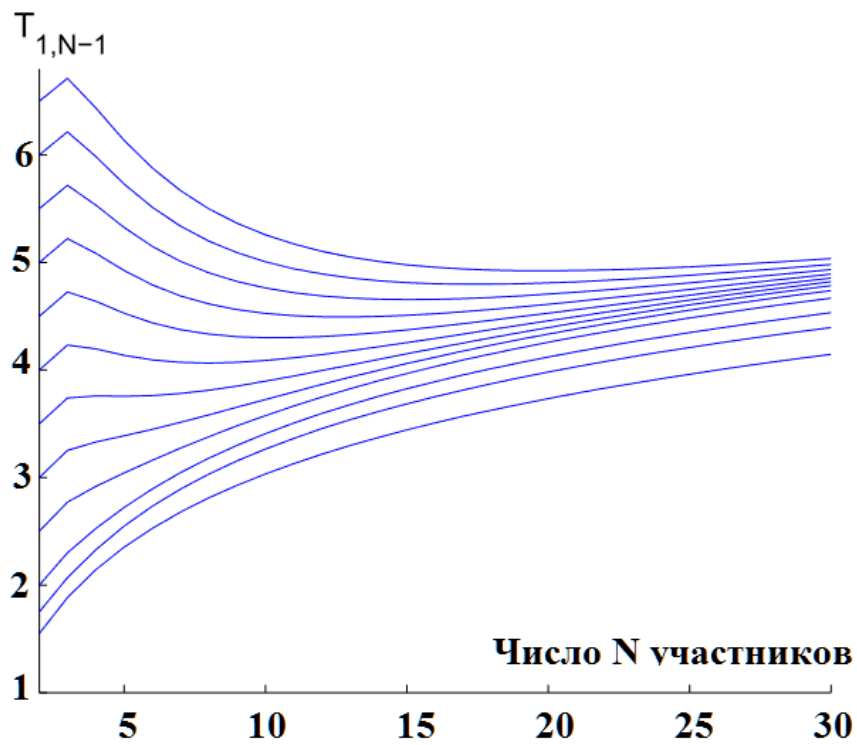


Рис. 2.3. $T_{1,N-1}$, $\lambda=1$. На верхнем рисунке $N \leq 30$ и $\mu=0,1, 0,5, 1, 2, \dots, 10$. На нижнем рисунке $N \leq 1000$, $\mu=0,1, 1, 10$; пунктирная линия показывает $H(N)$. Возрастающие значения μ соответствуют графикам снизу вверх

Увеличение N имеет два противоречивых эффекта: нужно

информировать больше агентов, но также больше агентов служит для распространения информации. Какой из эффектов преобладает, зависит от числа агентов и от ρ . Однако, для любого значения ρ (или μ на рисунках, где λ нормировано к единице) кривые всегда возрастают, когда N увеличивается с двух до трех (при увеличении μ рост менее значителен). Можно доказать правильность этого наблюдения следующим образом.

Утверждение 2.1. $T_{1,2}^{(N=3)} = T_{1,1}^{(N=2)}$

Доказательство. Доказательство применяет анализ пути. Рассмотрим процесс $G2$ с двумя агентами и трехагентный процесс $G3$. Агенты в $G2$ i (информированный агент) и u_1 (неосведомленный агент). Первоначально оба находятся в L . В $G3$, у нас изначально есть дополнительный неосведомленный агент u_2 в L . Рассмотрим первый переход i в M . Есть четыре возможных случая для агентов, с которым i встретится в M :

1. только u_1
2. оба u_1 и u_2
3. только u_2
4. никто

Во втором случае оба процесса завершаются. Случаи 1 и 3 имеют одинаковую вероятность. В случае 1 $G2$ завершается, тогда как в случае 3 $G3$ еще не завершено. Таким образом, пока $G2$ имеет более короткое ожидаемое остаточное время распространения. Теперь рассмотрим случай 4. Если следующим событием является возвращение i в L , то вернулись в исходную точку. В противном случае есть равная вероятность того, что следующим событием будет переход u_1 или u_2 в M . Опять же, в первом случае $G2$ завершается, тогда как в последнем случае $G3$ все еще продолжается. Вывод снова состоит в том, что $G2$ имеет меньшее ожидаемое время распространения.

Замечание 2.2. Видно, что время распространения увеличивается при увеличении числа агентов от двух до трех. Эффект дальнейшего

увеличения N зависит от μ . При малых значениях μ $T_{1,N-1}$ монотонно возрастает по N . При больших значениях μ функция сначала убывает, а затем возрастает. Для еще больших значений μ функция имеет вид монотонно убывающей. Значение μ при $T_{1,2}$ $T_{1,3}$ приблизительно равно 4.33.

2.1.2. Ожидаемое время для информирования выбранной части узлов

Предположим, что нас интересуют ожидаемые времена $T_{i,u}^{(\alpha)}$ и $\hat{T}_{i,u}^{(\alpha)}$, необходимые для информирования части α всей популяции. Тогда $T_{i,u}^{(\alpha)} = 0$, если число информированных агентов $N-u$ превышает $N\alpha$ или $u \leq N(1-\alpha)$. Аналогично, $\hat{T}_{i,u}^{(\alpha)} = 0$, если $i \geq \alpha N$. Пусть $N_0 = [(1-\alpha)N]$. Тогда, согласно замечанию 1, это в точности те состояния, которые находятся в первых N_0+1 столбцах диаграммы перехода рис. 2.2. Таким образом, единственное изменение в уравнениях баланса состоит в том, что начальные условия, заданные уравнением (2.1), расширяются таким образом, что все переменные, соответствующие первым столбцам N_0 на рис. 2.2, равны нулю. Для других столбцов уравнения (2.7) и (2.8) не меняются.

Замечание 2.3. Типичные примеры показаны на рис. 2.4. Мы видим, что ожидаемое время, необходимое для информирования дополнительного агента, сначала мало, а затем увеличивается. Таким образом, любое ослабление требования информировать все узлы имеет большое значение с точки зрения времени распространения. Например, когда $\mu=1$, оставление 5 из 100 агентов в неведении снижает ожидаемое время распространения с 5,76 до 3,46.

На рис. 2.5 представлены аналогичные графики, но с точки зрения доли целевых агентов в сравнении с долей времени, необходимого для достижения этой цели, в ожидании, относительно времени, необходимого

для информирования всех агентов.

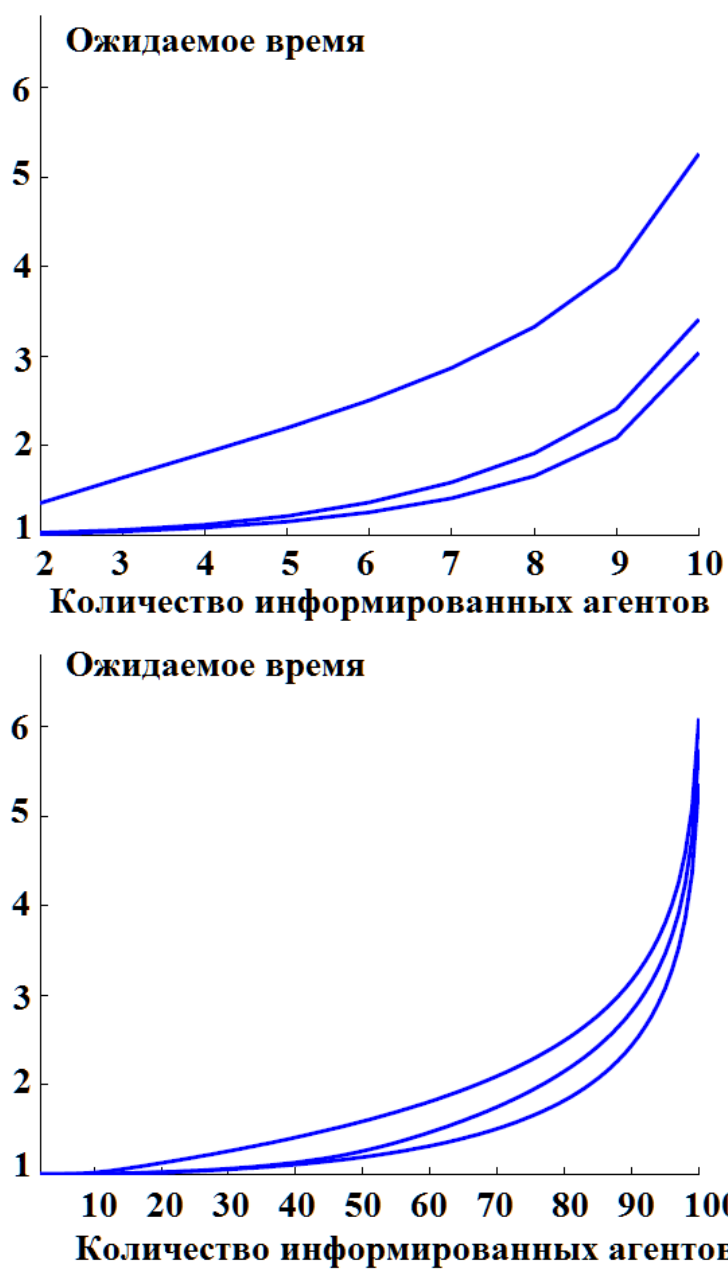


Рис. 2.4. Ожидаемое время до того, как будут проинформированы хотя бы $x=2, \dots, N$ агентов. Здесь $\lambda=1$. На каждом рисунке нижний график соответствует $\mu=0,1$, средний график соответствует $\mu=1$, а верхний график соответствует $\mu=10$. На верхнем рисунке $N=10$, а на нижнем рисунке $N=100$

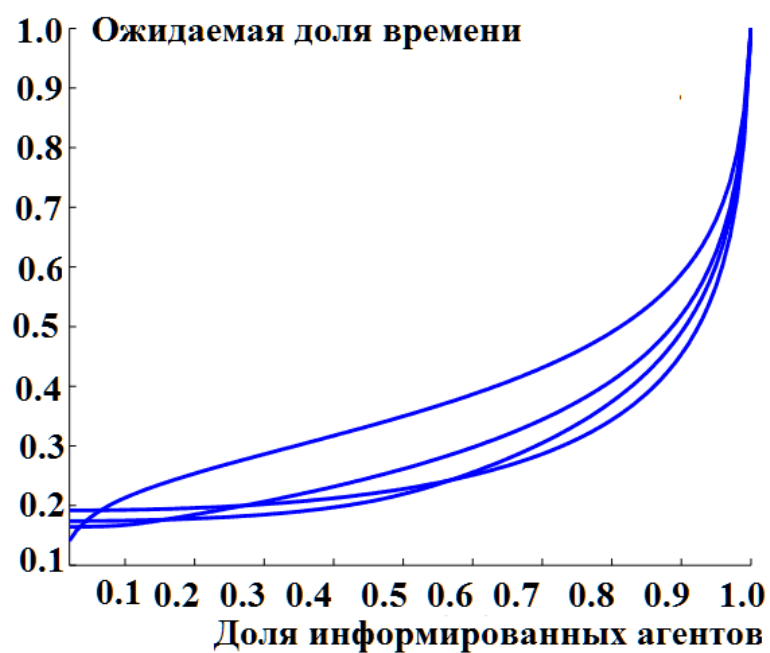
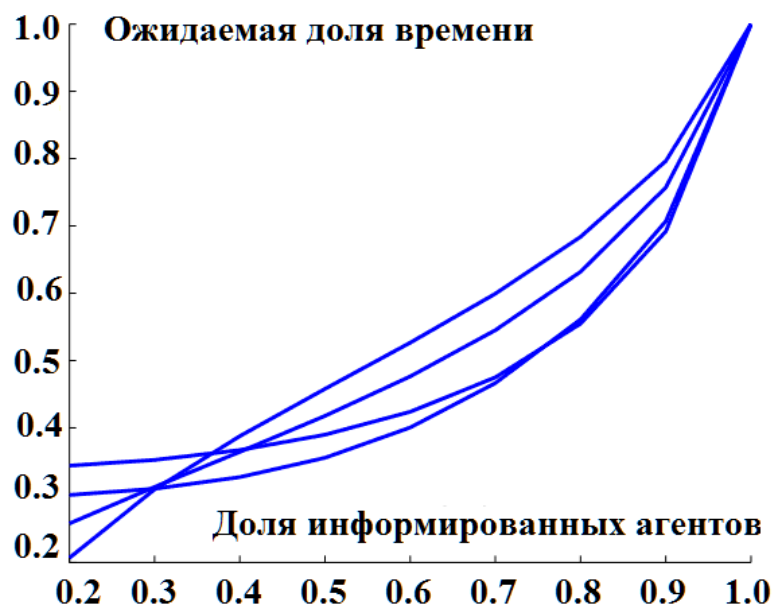


Рис. 2.5. Ожидаемая доля времени до того, как данная часть популяции будет проинформирована. На верхнем рисунке $N=10$, на нижнем $N=100$. На каждом рисунке графики соответствуют $\mu=0.01, 1, 10, 100$ сверху вниз около нуля и снизу вверх около единицы

Замечание 2.4. Видно, что кривые, связанные с меньшими значениями μ , меньше в нуле, но опережают кривые, связанные с большими значениями μ , и, таким образом, для достижения больших

долей населения требуется большее относительное время. Это отражает свойство, заключающееся в том, что при малом μ процесс вначале идет относительно быстрее, потому что существует более высокая вероятность встречи с информированным агентом в точке M .

2.2. Распространение во временном окне

Теперь рассмотрим информацию, которая актуальна только в пределах данного временного окна $[0, t]$, и после этого интервала агенты перестают распространять информацию. Интересует количество агентов, получивших информацию в течение временного окна, и вероятность того, что процесс завершится до t после того, как все агенты будут проинформированы.

2.2.1. Кумулятивная функция распределения $T_{1,N-1}$

Теперь рассмотрим кумулятивную функцию распределения (CDF) времени распространения. Для $t > 0$ мы вычисляем вероятность успеха; то есть процесс завершается (информируются все) не позднее t . Пусть $P(i, u, t)$ и $\hat{P}(i, u, t)$ обозначают вероятности того, что процесс завершится не позднее, чем через t единиц времени с этого момента, при заданном текущем состоянии (i, u) , агенты в M информированы и не информированы соответственно. Эти вероятности меняются вместе с t следующим образом.

Первоначально, $P(i, u, 0) = 1$ если $u = 0$ и $P(i, u, 0) = 0$ если $u > 0$. Аналогично, $\hat{P}(i, u, 0) = 0$ если $i < N$ и $\hat{P}(N, 0, 0) = P(N, 0, 0) = 1$. Также, для всех $t > 0$ и для всех i , $P(i, 0, t) = 1$.

Для $t > 0$ и $i + u < N$,

$$\begin{aligned}
P(i, u, t + dt) = & [1 - (i + u)\lambda dt - (N - i - u)\mu dt]P(i, u, t) + \\
& + i\lambda dt P(i - 1, u, t) \quad \text{при } i > 0 \\
& + u\lambda dt P(i, u - 1, t) \quad \text{при } u > 0 \\
& + (N - i - u)\mu dt P(i + 1, u, t) + o(dt)
\end{aligned}$$

$$\begin{aligned}
\hat{P}(i, u, t + dt) = & [1 - (i + u)\lambda dt - (N - i - u)\mu dt]\hat{P}(i, u, t) + \\
& + i\lambda dt \hat{P}(i - 1, u, t) \quad \text{при } i > 0 \\
& + u\lambda dt \hat{P}(i, u - 1, t) \quad \text{при } u > 0 \\
& + (N - i - u)\mu dt \hat{P}(i, u + 1, t) + o(dt)
\end{aligned}$$

Для $t > 0$, $i > 0$, $u > 0$, и $i + u = N$,

$P(N, 0, t) = 1$ для $t > 0$,

а при $i < N$

$$\begin{aligned}
P(i, u, t + dt) = \hat{P}(i, u, t + dt) = & (1 - N\lambda dt)P(i, u, t) + \\
& + i\lambda dt P(i - 1, u, t) + u\lambda dt \hat{P}(i, u - 1, t) + o(dt)
\end{aligned}$$

На рис. 2.6 показан типичный график вероятности $P(1, N-1, t)$ того, что все узлы проинформированы ко времени t (успешная вероятность). Заметим, что при $\rho \rightarrow \infty$ процесс завершается, когда все агенты переходят в М. Вероятность того, что это происходит до t , является просто вероятностью того, что данный агент переместится в М до того, как t возводится в степень N , то есть $(1 - e^{-\lambda t})^N$. Эта вероятность показана пунктирными кривыми на рис. 2.6 вместе с вычислением CDF для $\mu = 10$ и $\mu = 50$, когда $N \in \{5, 30\}$ и $\lambda = 1$.

Замечание 2.5. CDF имеет S-образную форму, аналогичную нормальному распределению. Действительно, $T_{1, N-1}$ представляет собой сумму случайных величин, соответствующих времени, необходимому для увеличения числа информированных клиентов на одного. Однако эти случайные величины не являются независимыми; например, несколько агентов часто информируются одновременно. Также можем видеть, что распределение не сходится к нормальному распределению по формуле

$$T_{1, N-1} \rightarrow (1 - e^{-\lambda t})^N \text{ при } \rho \rightarrow \infty.$$

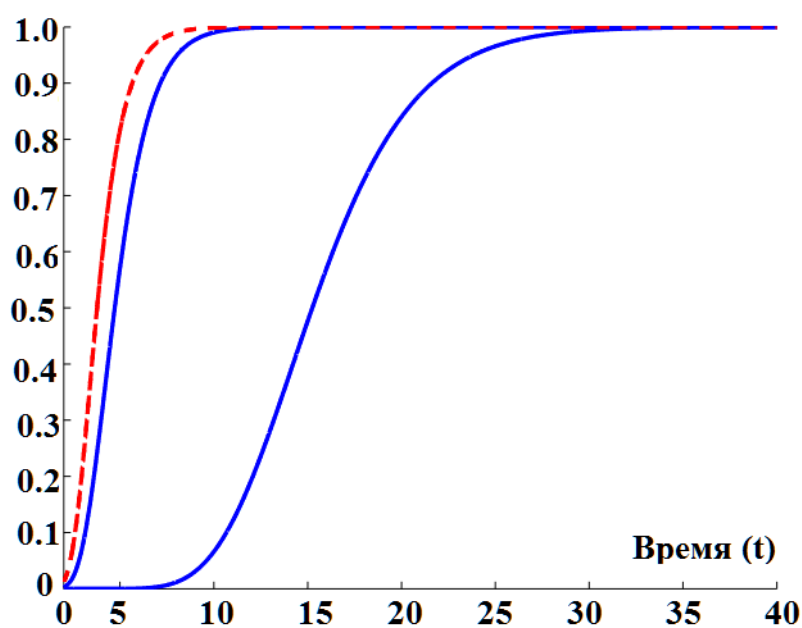
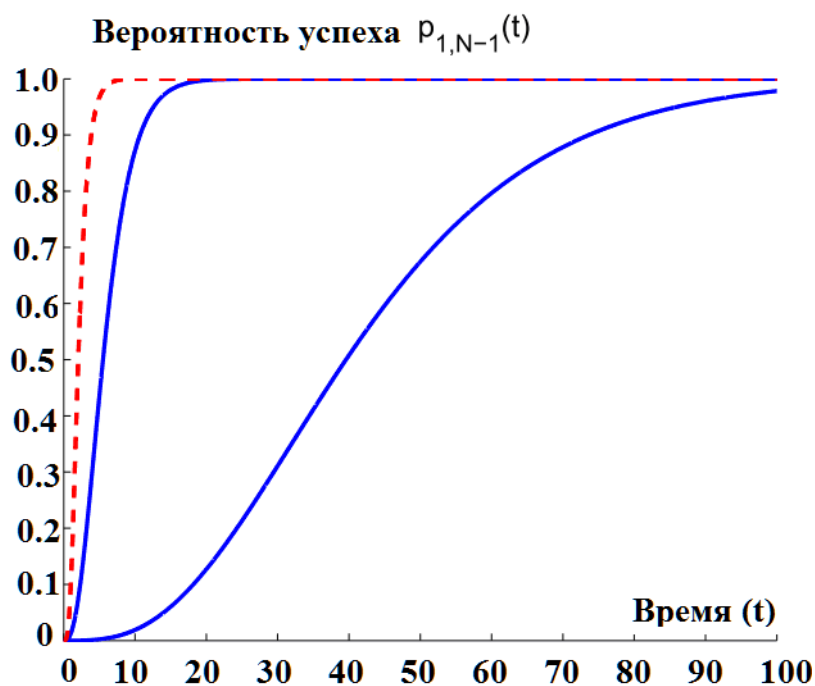


Рис. 2.6. Кумулятивные функции распределения для $T_{1,N1}$ при $\mu \rightarrow 0$ (верхний пунктирный график), $\mu=10$ (средний график), и $\mu=100$ (нижний график). Здесь $\lambda=1$. На верхнем рисунке $N=5$, на нижнем рисунке $N=30$

Замечание 2.6. На рис. 2.7 показана чувствительность CDF к изменению N . С увеличением N уменьшаются как вероятности очень короткого, так и очень большого времени распространения. Это можно

объяснить тем, что, с одной стороны, при увеличении N требуется информирование большего числа агентов, и вероятность того, что это будет выполнено за короткий промежуток времени, уменьшается.

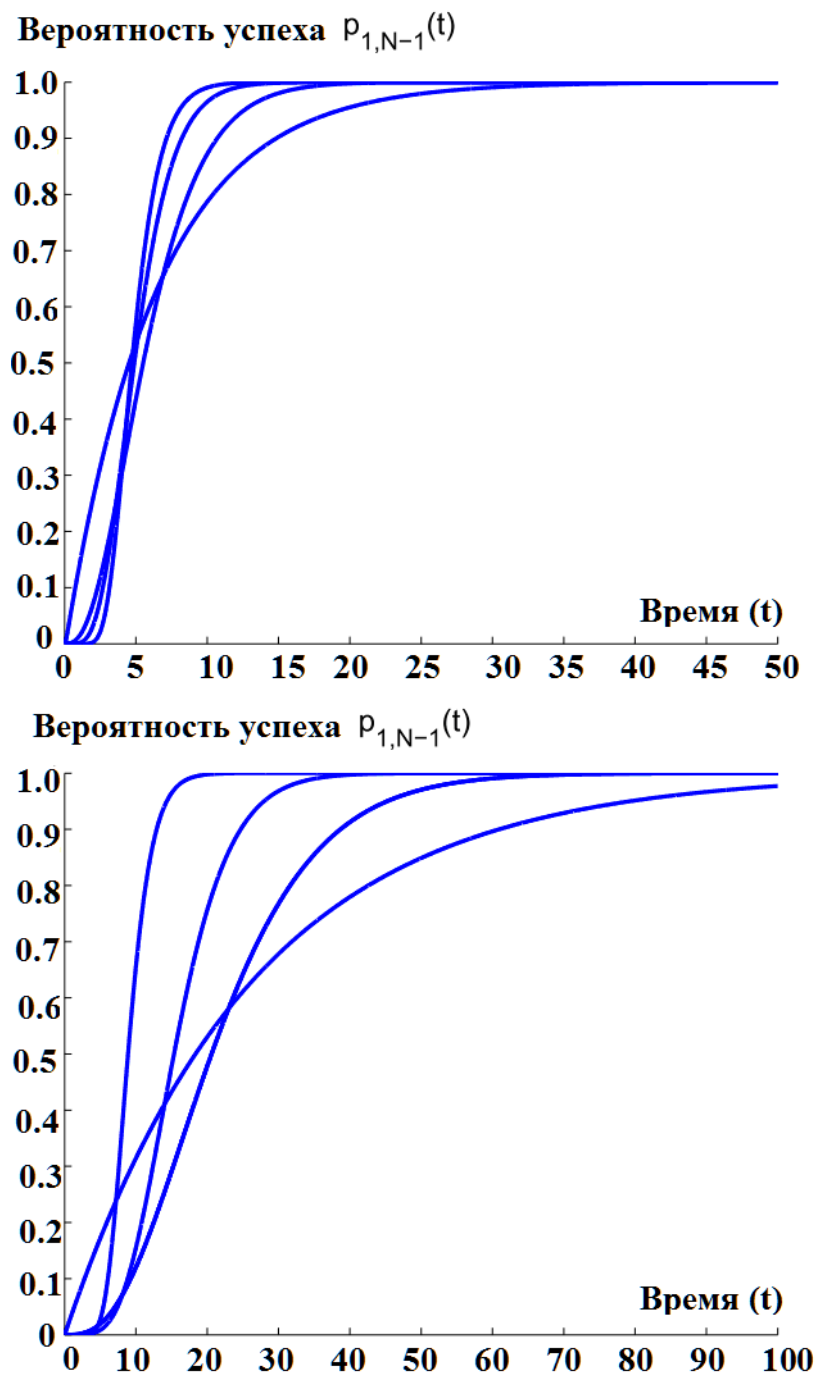


Рис. 2.7. Кумулятивные функции распределения для $T_{1,N}$ при $\mu=10$ (верх) и $\mu=50$ (низ). Здесь $\lambda=1$. Также $N=2, 5, 10, 30$, где N увеличивается сверху вниз вблизи $t=0$ и снизу вверх при больших значениях t .

С другой стороны, когда N увеличивается, быстро формируется зародыш информированных агентов, и с этого момента распространение происходит быстро.

2.2.2. Число информированных агентов

Сколько агентов будет проинформировано через время t после начала процесса? Теперь вычислим вероятности состояний в момент времени t и, в частности, ожидаемое число информированных агентов.

Это число легко вычислить при $\rho \rightarrow \infty$. В этом случае узел информируется в момент времени t , если либо он является изначально информированным агентом, либо и этот агент, и первоначально информированный переместились из L в M до времени t . Следовательно, ожидаемое количество информированных агентов в момент времени t равно $1 + (N-1)(1 - e^{-\lambda t})^2$.

Опишем общее решение. Пусть $Q(i, u, t)$ и $\hat{Q}(i, u, t)$ обозначают вероятности соответствующих состояний в момент времени t . Первоначально, когда $t=0$, $Q(1, N-1, 0)=1$ и, что эквивалентно, $\hat{Q}(1, N-1, 0)=1$, тогда как все остальные состояния в момент времени t имеют нулевую вероятность. Тогда мы имеем следующие дифференциальные уравнения (с точностью до погрешности $o(dt)$). Для $i+u < N$,

$$\begin{aligned} Q(i, u, t + dt) = & [1 - (i + u)\lambda dt - (N - i - u)\mu dt]Q(i, u, t) + \\ & + (N - i - u + 1)\mu dt Q(i - 1, u, t) \quad \text{при } i > 0 \\ & + (i + 1)\lambda dt Q(i + 1, u, t) \\ & + (u + 1)\lambda dt Q(i, u + 1, t) \quad \text{при } i + u + 1 < N \\ & + (i + 1)\lambda dt \hat{Q}(i + 1, u, t) \quad \text{при } i + u + 1 < N \end{aligned} \quad (2.*)$$

строгое неравенство в последней строке предотвращает дублирование с (*), когда $i+u+1=N$). Поэтому, для $i+u=N$,

$$\begin{aligned}\hat{Q}(i, u, t + dt) = & [1 - (i + u)\lambda dt - (N - i - u)\mu dt] \hat{Q}(i, u, t) + \\ & + (N - i - u + 1)\mu dt \hat{Q}(i, u - 1, t) \quad \text{при } u > 0 \\ & + (u + 1)\lambda dt \hat{Q}(i, u + 1, t)\end{aligned}$$

Вероятность того, что к моменту времени t будет проинформировано по крайней мере k агентов, равна

$$\sum_{\substack{u \leq N-k \\ i+u \leq N}} Q(i, u, t) + \sum_{\substack{i \geq k \\ i+u < N}} \hat{Q}(i, u, t)$$

Вероятности $P(i, u, t)$ и $\hat{P}(i, u, t)$ связаны с вероятностями $Q(i, u, t)$ и $\hat{Q}(i, u, t)$. Например, $P(1, N-1, t)$ может быть вычислено как $\sum_{i=0}^N Q(i, 0, t)$.

Ожидаемое число информированных агентов

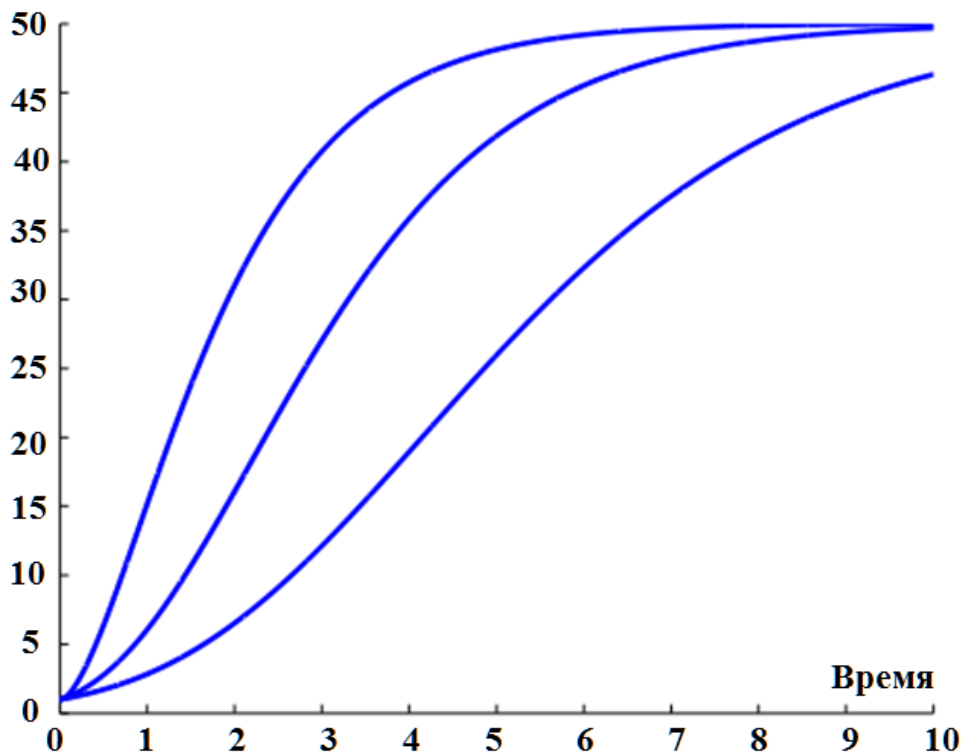


Рис. 2.8. Ожидаемое количество информированных агентов. Здесь $\lambda=1$. Сверху вниз, $N=50$ и $\mu=10, 50, 100$.

На рис. 2.8 показано поведение ожидаемого числа информированных агентов в зависимости от времени. Кривые имеют S-образную форму,

аналогичную общепринятым эмпирическим данным и теоретическим моделям.

2.3. Управление процессом

2.3.1. Зависимость времени распространения от интенсивности контактов

Если исследуется распространение информации, одной из задач является ускорение процесса распространения. Напротив если речь идет о распространении вредоносной информации, мы будем заинтересованы в ее замедлении. Естественной регулируемой переменной является интенсивность контакта ρ , которой можно управлять, изменяя λ или μ . Рис. 2.9 показывает зависимость ожидаемого времени распространения от μ для различных размеров популяции и $\lambda=1$.

Замечание 2.7. Интересно, что кривые пересекаются. Это можно увидеть из формул выше, которые для небольшого μ (высокого ρ) времени распространения, $H(N)/\lambda$ увеличивается с N , поэтому процесс быстрее, когда N небольшой. Напротив, когда μ является большим (ρ мало), время распространения $(\mu/\lambda^2)H(N-1)/N$ уменьшается с N (для $N>2$).

2.3.2. Достижение целевого времени распространения

Рис. 2.10 показывает количество мест встречи, необходимое для достижения заданного $T_{1,N-1}$, как функции от N где $\lambda=1$. Можно явно вычислить пересечения кривых с осями. Пересечение с осью μ при $N=2$, с $\lambda=1$, $T_{1,1}=1.5+\mu/2$, $\mu=2T_{1,1}-3$. Пересечение с осью N при $\mu=0$, тогда целевое время распространения равно сумме гармоник $H(N)$. Например, как видно на рис. 10, $H(30) \approx 4$; следовательно нижняя кривая которая соответствует $T_{1,N-1}=4$, пересечет $\mu=0$ около $N=30$. Аналогично, кривая выше, которая соответствует $T_{1,N-1}=4.5$, пересечет 0 в $H(50) \approx 4.5$.

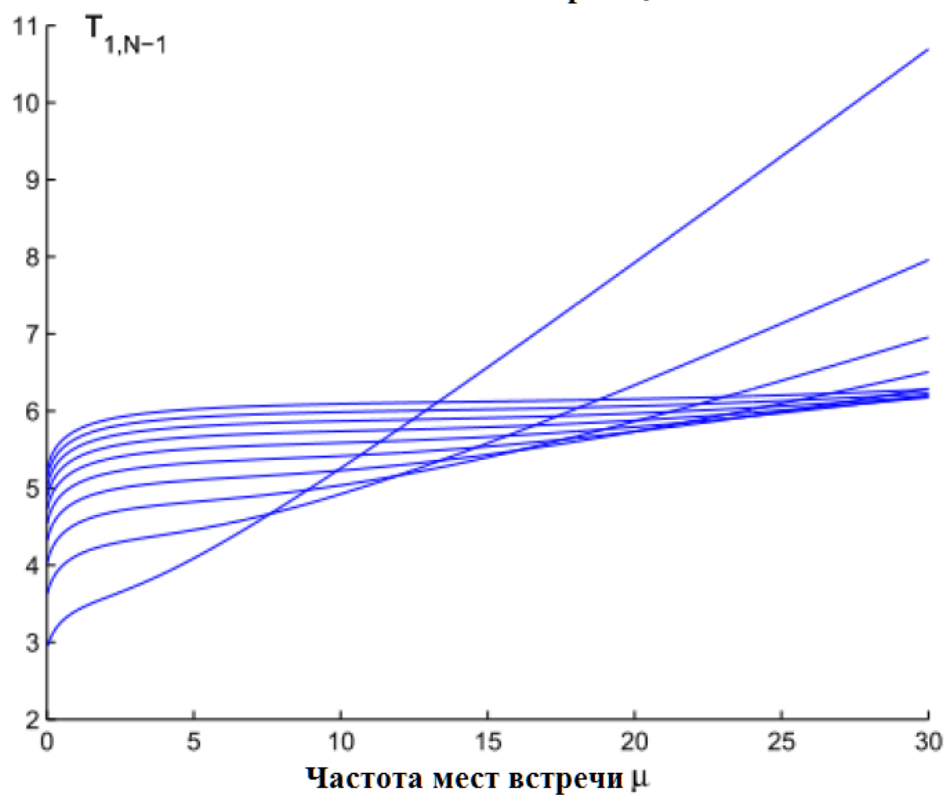
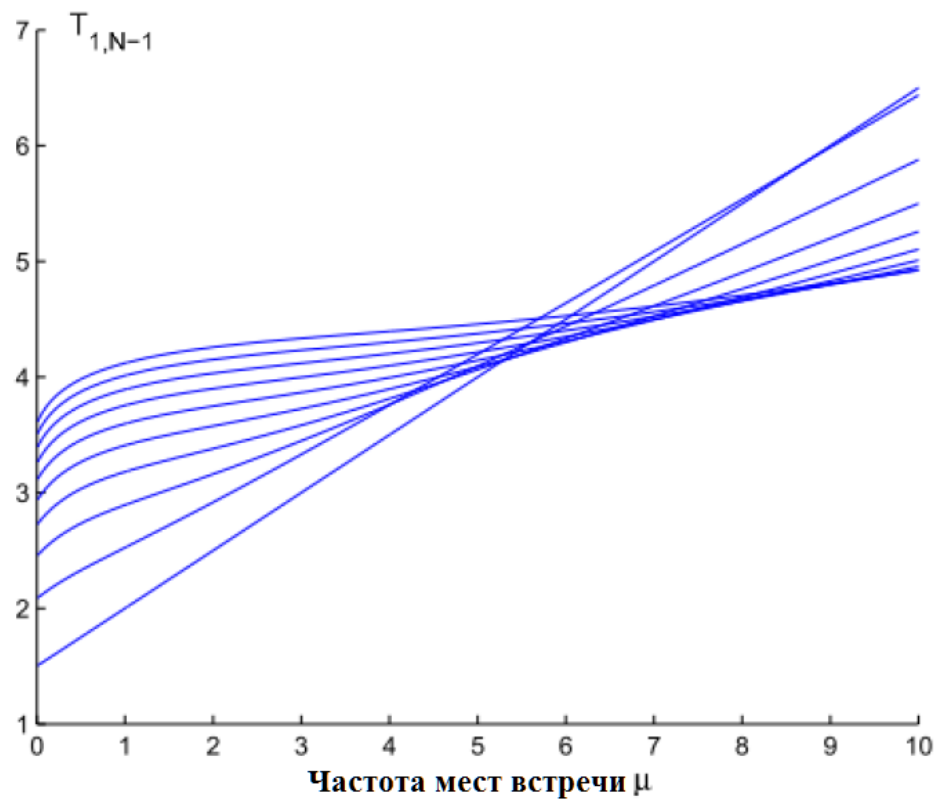


Рис. 2.9. $T_{1,N-1}$ как функция частоты мест встречи μ . Здесь $\lambda=1$. Также $N=2, 4, \dots, 20$ (верх) и $N=10, 20, \dots, 100$ (низ). Возрастающие значения N прикрепляются к графикам снизу вверх вблизи нуля и сверху вниз для больших значений μ

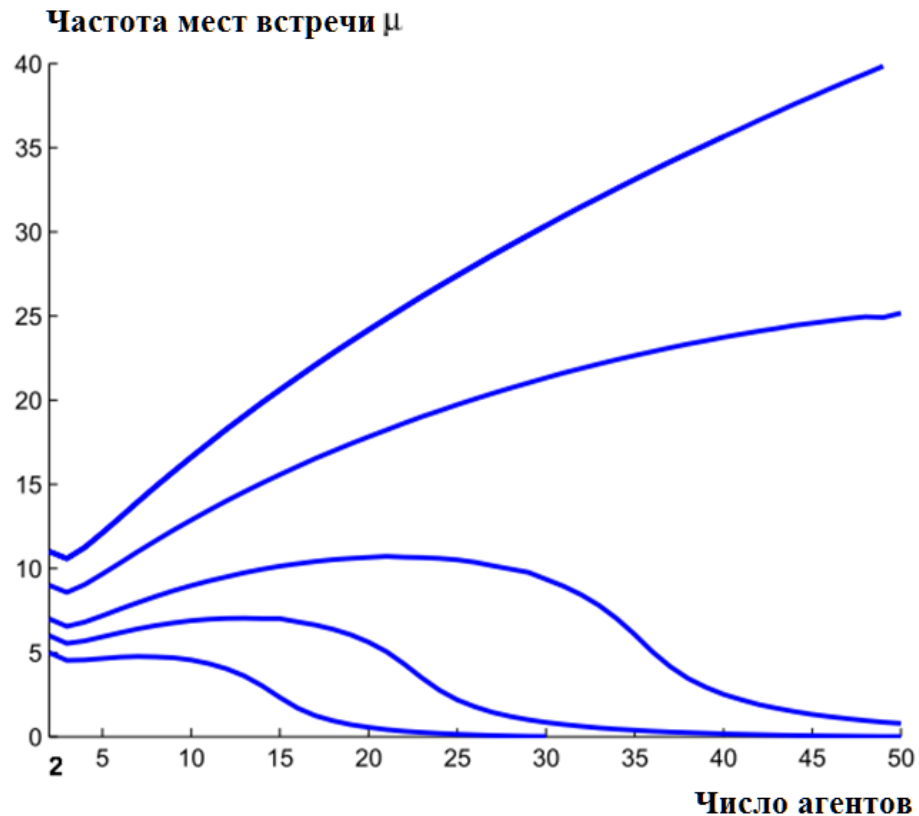


Рис. 2.10. Частота мест встречи μ , необходимая для достижения заданного ожидаемого времени распространения $T_{1,N-1}$, в зависимости от числа агентов N , для $T_{1,N-1}=4, 4,5, 5, 6, 7$ и $\lambda=1$. Возрастающие значения T соответствуют графикам снизу вверх

Замечание 2.8. Кривые сначала уменьшаются при увеличении числа агентов с двух до трех, а затем растут, в конце концов уменьшаясь, пока не достигают нуля, где $H(N) \approx T_{1,N-1}$. Таким образом, когда N велико, нам нужно малое μ для достижения заданного целевого времени распространения. Но и когда N мало, нам нужно относительно малое μ .

2.3.3. Выбор начального количества источников

В разделе 2.2.3 рассматривался процесс, который завершается при достижении заданного числа информированных агентов. Напротив, в этом разделе предполагаем, что процесс начинается с заданного числа информированных агентов.

Предположим, что начинаем с начального числа из k информированных агентов. Теперь исследуем влияние размера набора источников на время распространения.

Когда $\rho \rightarrow 0$, легко дополнить вывод, ранее. Учитывая начальное число размера k , ожидаемое время, пока все агенты не будут проинформированы (аналогично уравнению (2.6)):

$$\sum_{i=k}^{N-1} \frac{1}{N} \left(\frac{1}{N-i} + \frac{1}{i} \right) = \frac{1}{N} \left(\sum_{j=1}^{N-k} \frac{1}{j} + \sum_{i=1}^{N-1} \frac{1}{i} \right) = \frac{1}{N} (H(N-k) + H(N-1)) \quad (2.6^*)$$

На рис. 2.11 показано ожидаемое время размножения для $N=30$ и $N=50$ и всех возможных размеров источников $k=1, \dots, N$.

Замечание 2.9. Из рис. 2.11 видно, что увеличение размера источников оказывает сильное влияние на сокращение времени размножения, когда размер либо очень маленький, либо очень большой, а промежуточный эффект весьма умеренный.

2.4. Выводы

1. Рассматривается простая модель распространения с мобильными агентами в звездообразной сети, которая может быть сведена к сети с двумя узлами. Тем не менее, анализ далеко не однозначен, и результаты не всегда легко предсказать. Компактное представление диаграммы перехода позволяет эффективно вычислять ожидаемое время распространения.

2. Представлены аналитические границы для крайних случаев, которые дополняются численными результатами и анализом чувствительности. Например, увеличение размера популяции имеет два противоречивых эффекта, которые заставляют время распространения сначала увеличиваться, затем уменьшаться и в конечном итоге увеличиваться с асимптотическим поведением, подобным гармонической сумме.

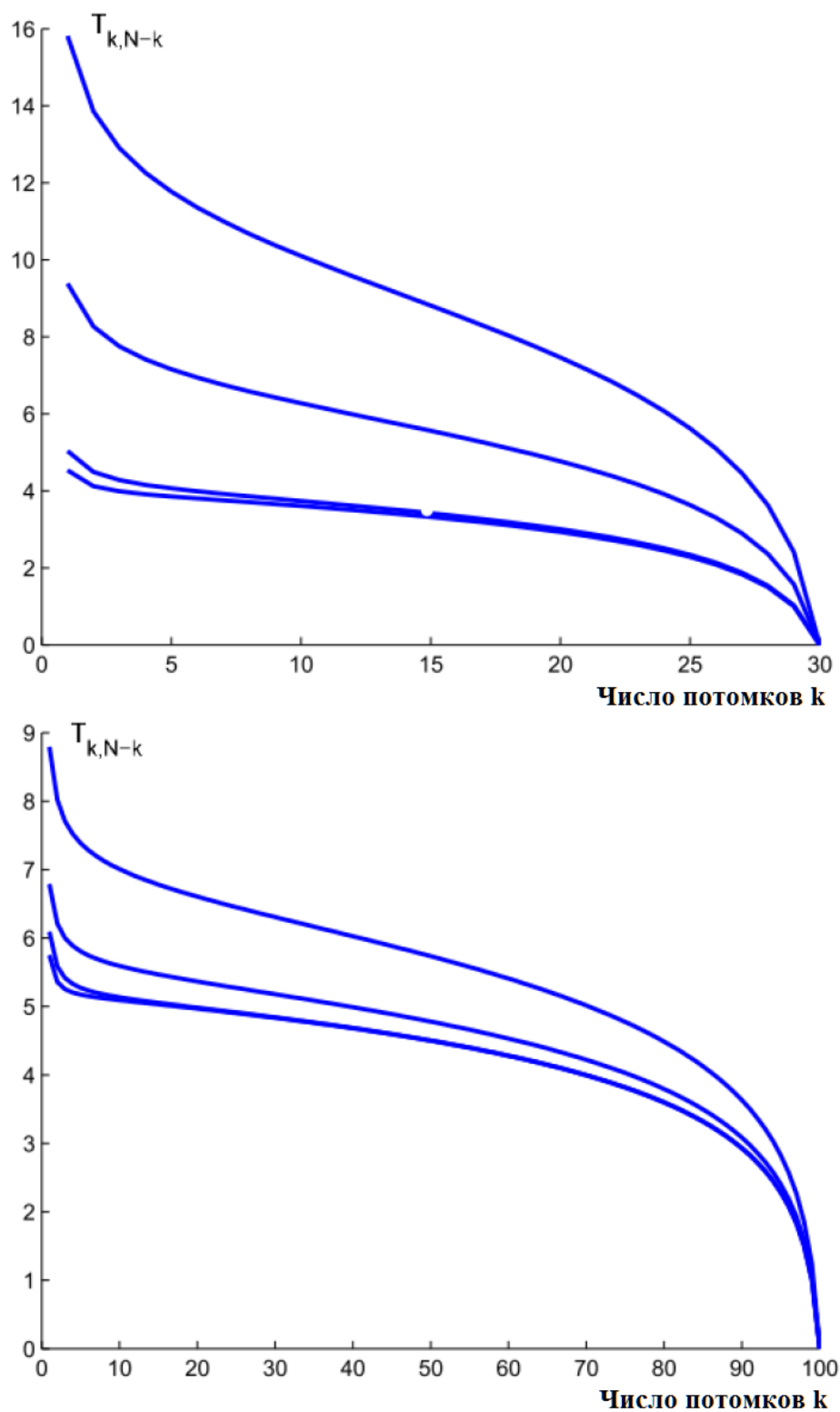


Рис. 2.11. Ожидаемое время распространения $T_{k,N-k}$ в зависимости от азмера начального числа k . На верхнем рисунке $N=30$, а на нижнем рисунке $N=100$. Также $\mu=1, 10, 50, 100$ с возрастающими значениями снизу вверх, а $\lambda=1$

В связи с этим ожидаемое время, необходимое для информирования дополнительного агента, сначала мало, а затем увеличивается, а вероятность информирования всех агентов в течение заданного срока имеет S-образную форму. Также предоставлена информация о том, как на процесс может повлиять изменение параметров моделирования, таких как начальное и конечное количество информированных агентов и интенсивность контактов.

3. Основная особенность модели, которая отличается от традиционных моделей процессов распространения, заключается в том, что мы допускаем одновременное взаимодействие узлов (передачу информации), тогда как общепринятое соглашение предполагает единичные взаимодействия. Следовательно, наша модель не определяет процесс рождения-смерти.

4. Представлен эффективный способ нахождения ожидаемого времени распространения, который вытекает из интересного представления диаграммы переходов состояний. Вместо решения системы линейных уравнений $O(N^2)$ достаточно решить $N-1$ систему из N уравнений, что явно является улучшением, если для решения уравнений не существует алгоритма линейной сложности.

5. Создана марковская модель распространения информации мобильными агентами в звездообразной сети, отличающаяся учетом информированности агентов и возможностью описания одновременной передачи информации между агентами и обеспечивающая уменьшение вычислительной сложности расчета ожидаемого времени распространения с $O(N^3)$ до $O(N^2)$.

Список литературы к главе 2

1. Centola D. The spread of behavior in an online social network// Science, 2010, 329(5996):1194-1197.
2. Colizza V., Vespignani A. Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: Theory and simulations// J. Theoret. Biol., 2008, 251(3):450-467.
3. De Arruda G.F., Rodrigues F.A., Rodriguez P., Cozzo E., Moreno Y. A general Markov chain approach for disease and rumour spreading in complex networks// J. Complex Networks, 2018, 6(2):215-242.
4. Feola G., Butt A. The diffusion of grassroots innovations for sustainability in Italy and Great Britain: An exploratory spatial data analysis// Geographical J., 2017, 183(1):16-33.
5. Chierichetti F., Lattanzi S., Panconesi A. Rumor spreading in social networks// Theoret. Comput. Sci., 2011, 412(24):2602-2610.
6. Gilboa-Freedman G., Hassin R. When Markov chains meet: A continuous-time model of network evolution// Statist. Probability Lett., 2016, 116:131-138.
7. Groeneveldt R., Nain P., Koole G. The message delay in mobile ad hoc networks// Performance Evaluation, 2005, 62:210-228.
8. Isella L., Stehlé J., Barrat A., Cattuto C., Pinton J.-F. What's in a crowd? Analysis of face-to-face behavioral networks// J. Theoret. Biol., 2011, 71(1):166-180.
9. Jin M, Liu F, Zhou C Rumor spreading: A survey// 2nd Int. Conf. on Artificial Intelligence Engrg. Appl., 2017, 263-269.
10. Manzo G., Gabbriellini S., Roux V., M'Mbogori F.N. Complex contagions and the diffusion of innovations: evidence from a small-N study// J. Archaeological Method Theory, 2018, 25(4):1109-1154.
11. Mascia M.B., Mills M. When conservation goes viral: The diffusion of innovative biodiversity conservation policies and practices// Conservation

Lett., 2018, 11(3):e12442, <https://doi.org/10.1111/conl.12442>.

12. Milling C., Caramanis C., Mannor S., Shakkottai S. Distinguishing infections on different graph topologies// IEEE Trans. Inform. Theory, 2015, 61(6):3100-3120.

13. Nowzari C., Preciado V.M., Pappas G.J. Analysis and control of epidemics// IEEE Control Systems Magazine, 2016, 36(1):26-46.

14. Pastor-Satorras R., Castellano C., Van Mieghem P., Vespignani A. Epidemic processes in complex networks// Rev. Modern Phys., 2015, 87(3):925-979.

15. Rogers E.M. Diffusion of Innovations, 4th ed. - Free Press, New York, 1995.

16. Sattenspiel L., Dietz K. A structured epidemic model incorporating geographic mobility among regions// Math. Biosciences, 1995, 128(1-2):71-91.

17. Simpson G., Clifton J. Testing diffusion of innovations theory with data: Financial incentives, early adopters, and distributed solar energy in Australia// Energy Res. Soc. Sci., 2017, 29:12-22.

18. Sun H., Cheng R., Xiao X., Yan J., Zheng Y., Qian Y. Maximizing social influence for the awareness threshold model// Database Systems for Advanced Applications, Lecture Notes in Computer Science, 2018, vol. 10827 (Springer, Cham), 491-510.

19. Vespignani A. Modelling dynamical processes in complex socio-technical systems// Nature Phys., 2012, 8:32-39.

20. Xiong H., Wang P., Bobashev G. Multiple peer effects in the diffusion of innovations on social networks: a simulation study// J. Innovation Entrepreneurship, 2018, 7(1):1-18.

21. Zhang X., Neglia G., Kurose J., Towsley D. Performance modeling of epidemic routing// Comput. Networks, 2007, 51(10):2867-2891

3. Мобильные программные агенты как инструмент управления нагрузкой в распределенных инфокоммуникационных системах

3.1. Мобильные агенты и балансировка нагрузки

Технология мобильных агентов обеспечивает поддержку балансировки нагрузки с тремя основными характеристиками:

- Они могут перемещаться с одной платформы на другую.
- Они могут перемещаться между платформами разнородного характера (например, операционная система, емкость процессора, хранилище и т.д.).
- Они содержат код конкретного приложения, вместо того чтобы требовать предварительной установки этого кода на целевой машине.

Два основных преимущества, которые дает подход мобильного агента для балансировки нагрузки, заключаются в следующем:

- сокращение сетевого трафика;
- миграция процессов с одного сайта на другой.

3.1.1. Предлагаемая архитектура

В этом разделе предложен вклад в балансировку нагрузки на основе парадигмы мобильного агента.

В дальнейшем предполагаем, что узлы находятся в «кооперативной» сети, что позволяет нам изучать только характеристики рассматриваемого алгоритма, и предполагается, что задачи независимы, т.е. между ними нет связи.

Для определения состояния каждой машины необходимо определить пороговые значения: существуют статические пороговые значения, которые являются неизменяемыми фиксированными значениями, и другие динамические пороговые значения, которые изменяются в зависимости от

эволюции системы. Используется второй вариант, в котором пороговое значение определяется в соответствии со средней нагрузкой системы, что больше подходит для динамической среды.

Предложены три типа агентов: стационарного агента и двух мобильных агентов (рис. 3.1):

- Агент-наблюдатель (ОА).
- Мобильный агент-супервайзер (SMA).
- Мобильный агент-транспортёр (ТМА).

Далее представлено изложение преимуществ каждого предлагаемого процесса в предложенном подходе, показывая различные случаи, которые могут возникнуть.

3.1.2. Исследование преимуществ предлагаемого процесса

1. ОА_i видит недогруженность собственного сайта. Отправляется волновое сообщение о состоянии, и перегруженный ОА_j информирует ОА_i об этом. Тогда возможны варианты:

- а. ОА_i догружается от ОА_k (запрос не принимается).
- б. Дополнительная нагрузка ОА_j перегрузит ОА_i (запрос не принимается).
- с. ОА_i догружается от ОА_j (запрос не принимается).

2. ТМА перемещается между узлами от S_j в поисках узла для перенаправления загрузки и находит нужный узел S_z.

3. ОА_j просто выбирает ОА_z с глобально минимальной загрузкой.

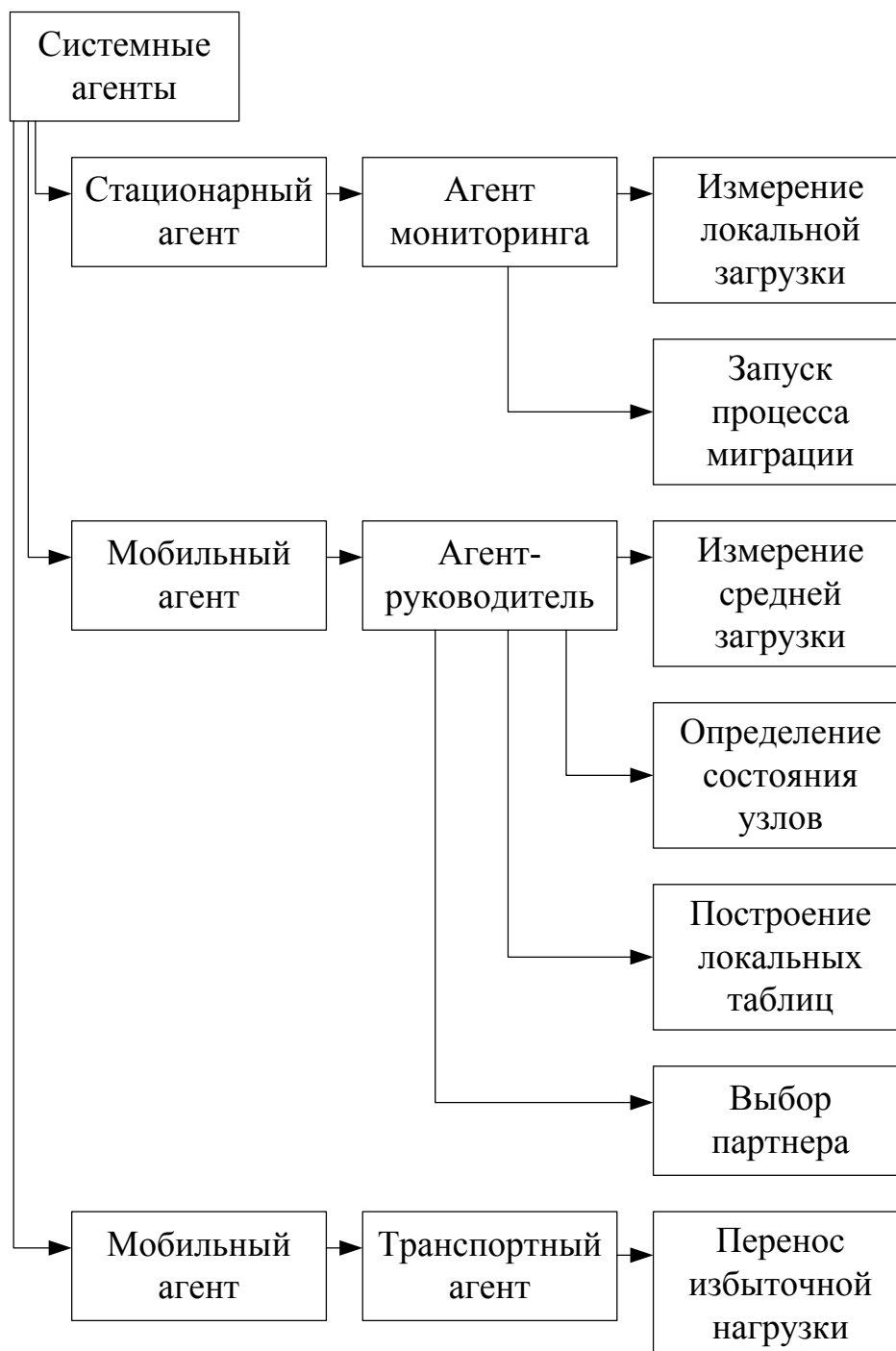


Рис. 3.1. Роль каждого предложенного агента

Описание предлагаемых агентов

Агент-наблюдатель (ОА)

Цель нашего алгоритма состоит в том, чтобы привести систему в сбалансированное состояние и, следовательно, иметь сбалансированные машины. Роль ОА заключается в наблюдении за нагрузкой машины, на

которой она расположена:

а. Измерение локальной нагрузки узла

Существует два метода измерения нагрузки: один - по требованию, а другой - периодический:

- По требованию: в этом случае ОА измеряет нагрузку на узел, когда SMA просит его сделать это (т. Е. когда

- SMA поступает на этот узел). Таким образом, ОА должен знать время прибытия SMA, что означает, что SMA должен сообщить ОА сигнальным сообщением. В противном случае SMA должен подождать, пока ОА не вычислит локальную нагрузку, что увеличивает время сбора нагрузки SMA.

- Периодический: в этом случае ОА не нужно знать время прибытия SMA, так как он измеряет локальную загрузку своего сайта каждый период Tinfo. Когда SMS приходит на этот сайт, ОА предоставляет ему последнее измеренное значение нагрузки. Информация должна быть указана за соответствующий период, чтобы избежать перегрузки системы. Если нагрузки быстро меняются, то информация должна быть небольшой. В противном случае необходимо увеличить продолжительность Tinfo.

Будем использовать второй метод, потому что он прост в реализации, не требует удаленного обмена сообщениями между SMA и ОА и сокращает время, выделяемое на политику сбора информации.

б. Запуск процесса миграции

Когда ОА сайта S_j получает сообщение от SMA, он получает сообщение о том, что его сайт перегружен. Поэтому этот ОА запускает агента TMS, указывая излишек для транспортировки, используя среднюю нагрузку, отправленную по SMS. Кроме того, конечный сайт объявлен в сообщении SMA.

Такая передача нагрузки между двумя несбалансированными машинами приближает их нагрузку к средней нагрузке системы, чтобы

сделать их более сбалансированными.

Алгоритм 3.1: Сценарий агента-наблюдателя

```
For each site Sk {  
  For each period Tinfo {  
    Measure local load Lk //Lk is the Local load of site Sk  
  }  
  Receive_info(SMA, Sr, AL); //SMA indicates the recipient site Sr.  
  //AL is the average load  
  If (Received_info != NULL) {  
    Launch the TMA agent and send it to the site Sr  
  } Else //Sk is not overloaded  
    OA declares: "My site is lightly loaded"  
    OA declares: "I am waiting for the reception of transporter agents from  
other sites"  
  }  
}
```

Мобильный агент-супервайзер (SMA)

SMA циклически перемещается между различными узлами сети. Для каждого сайта SMA извлекает имя сайта и его значение загрузки и помещает их в таблицу (tab_system). Закончив это движение, SMA делает два следующих шага.

а. Расчет средней нагрузки системы (AL)

По прибытии агента SMA на сайт он связывается с ОА, чтобы получить локальную нагрузку. Затем SMA добавляет это значение нагрузки к уже собранным значениям. Когда SMA завершает свою траекторию, он вычисляет среднюю нагрузку "AL".

Эта AL обеспечивает:

- Достижение глобальной балансировки нагрузки, поскольку нагрузка на каждую машину сравнивается с общей нагрузкой системы.
- Управление изменениями нагрузки, которые могут возникнуть в системе, чего не происходит при определении фиксированного порога.

б. Определение состояния узла и построение двух таблиц

Используем среднюю нагрузку (AL) в качестве "порога", в

соответствии с которым мы можем различать 3 состояния:

- сайт перегружен, если его загрузка $> AL$;
- сайт недогружен, если его загрузка $< AL$;
- нейтральное состояние, если его нагрузка $= AL$.

Из таблицы Tab_System и рассчитанной средней нагрузки агент SMA строит две подтаблицы:

- Без вкладок: содержит недогруженные сайты.
- Tab_plus: содержит перегруженные сайты.

После этого SMA упорядочит значения нагрузки для каждой из этих таблиц:

- Без табуляции: в порядке возрастания (от наименьшего значения нагрузки до наибольшего).
- Tab_Plus: в порядке убывания.

в. Выбор партнера

После этих двух шагов создается глобальное видение общей картины, и теперь SMA играет роль планировщика. Он определяет для каждого перегруженного сайта партнера (недогруженный сайт получателя) таким образом, чтобы перегруженный сайт и его партнер имели одинаковый индекс таблицы соответственно в tab_plus и tab_less.

В подробном виде предложенный для выбора партнера метод состоит в том, чтобы назначить первую запись Tab_Less первой записи Tab_Plus. Это делается для того, чтобы назначить сайт с наибольшей нагрузкой сайту с наименьшей нагрузкой и назначить вторую запись Tab_Less второй записи Tab_Plus и т.д. - каждый недогруженный сайт Tab_Less [i] является партнером перегруженного сайта Tab_Plus [i].

Мобильный агент должен быть быстрым во время своего перемещения, чтобы информация о загрузке системы не изменялась во время этого перемещения, т.е. агент должен быть обновлен, таким образом, размер кода SMA должен быть сведен к минимуму.

В предложенной архитектуре можно использовать один агент-супервизор, но всегда можем рассмотреть возможность увеличения числа этих агентов, и это зависит от размера сети (количества участвующих машин). В этом случае необходимо обеспечить механизм сотрудничества между различными агентами-супервизорами, чтобы сформировать глобальное видение нагрузки на систему.

Алгоритм 3.2: скрипт мобильного агента-супервизора

```

Som=0 ; AL=0 ;
For m = (from 1 to nbSites) { // nbSites is the number of machines or
nodes
  Request the local load from the OA of site Sj
  Add(site name Sj + Lj load) as an entry in tab_system
  Som=Som+Lj // Lj: local load provided by agent OA of site Sj
  doMove (Sj, Sk) // SMA moves to the next node
}
AL = Som / nbSites ;
While Tab_Sys[i] not_empty {
  If (Tab_System [i] > AL)
  Add this site as an entry to tab_plus
  Else {
  If (Tab_Systeme [i] < AL)
  Add this site as an entry to tab_less
  }
  i++; //next hut in tab_sys
}
Ordering Tab_Less in ascending order;
Ordering Tab_Plus in descending order;
While tab_plus [i] not_empty { //indicating partners
  So = Tab_Plus[i] // source overloaded site
  Su = Tab_Moins[i] //receiver underloaded site
  Send_info (Sk, Sj ,AL) // SMA informs the overload site by the receiver
of its surplus load
}

```

Мобильный агент-транспортёр (ТМА)

Это мобильный агент, запущенный агентом ОА перегруженного сайта, и он представляет собой избыток этого последнего. Целью ТМА является транспортировка этого излишка на сайт получателя, на котором

будет выполняться задача (задачи).

Это агент ОА, который указывает ТМА на работу, которую нужно выполнить. Эта работа представляет собой набор задач. Количество агентов ТМА зависит от того, сколько задач они могут выполнить.

Алгоритм 3.3: сценарий мобильного агента-транспортёра

doMove(Sk,Sj) ;

// TMA moves from the overloaded site to the receiver site

Send (L_Plusk, Sk, Sj) ; // Sk : the source site name

// L_Plusk : excess load of site Sk

// Sj: the recipient site name (indicated in SMA script)

Обзор предложенной архитектуры системы со сценарием трех агентов на 4 сайтах представлен на рис. 3.2.

3.2. Результаты эксперимента

3.2.1. Состояние нагрузок

Для тестирования поведения и оценки эффективности предлагаемого подхода алгоритмы агентов разрабатываются с использованием Java Eclipse IDE поверх платформы агента JADE. Используется "FSMBehaviour" для агентов ОА и SMA. Такое поведение служит для представления сложных задач, оно отвечает потребностям сложного поведения этих двух агентов. Для третьего агента ТМА, у которого есть только одна задача для выполнения, простое поведение удовлетворяет потребности, поэтому в данном случае это "OneShotBehaviour". Исходный код состоит из трех классов и их подклассов в соответствии с ролями на рис. 3.1. Чтобы запустить выполнение разработанного алгоритма в Jade, сначала запускаются агенты-наблюдатели (реализовано восемь агентов ОА0, ОА1, ...), каждый в контейнере. Затем запускается агент SMA в главном контейнере. Затем агент АМТ запускается соответствующим агентом ОА (агентом перегруженного сайта).

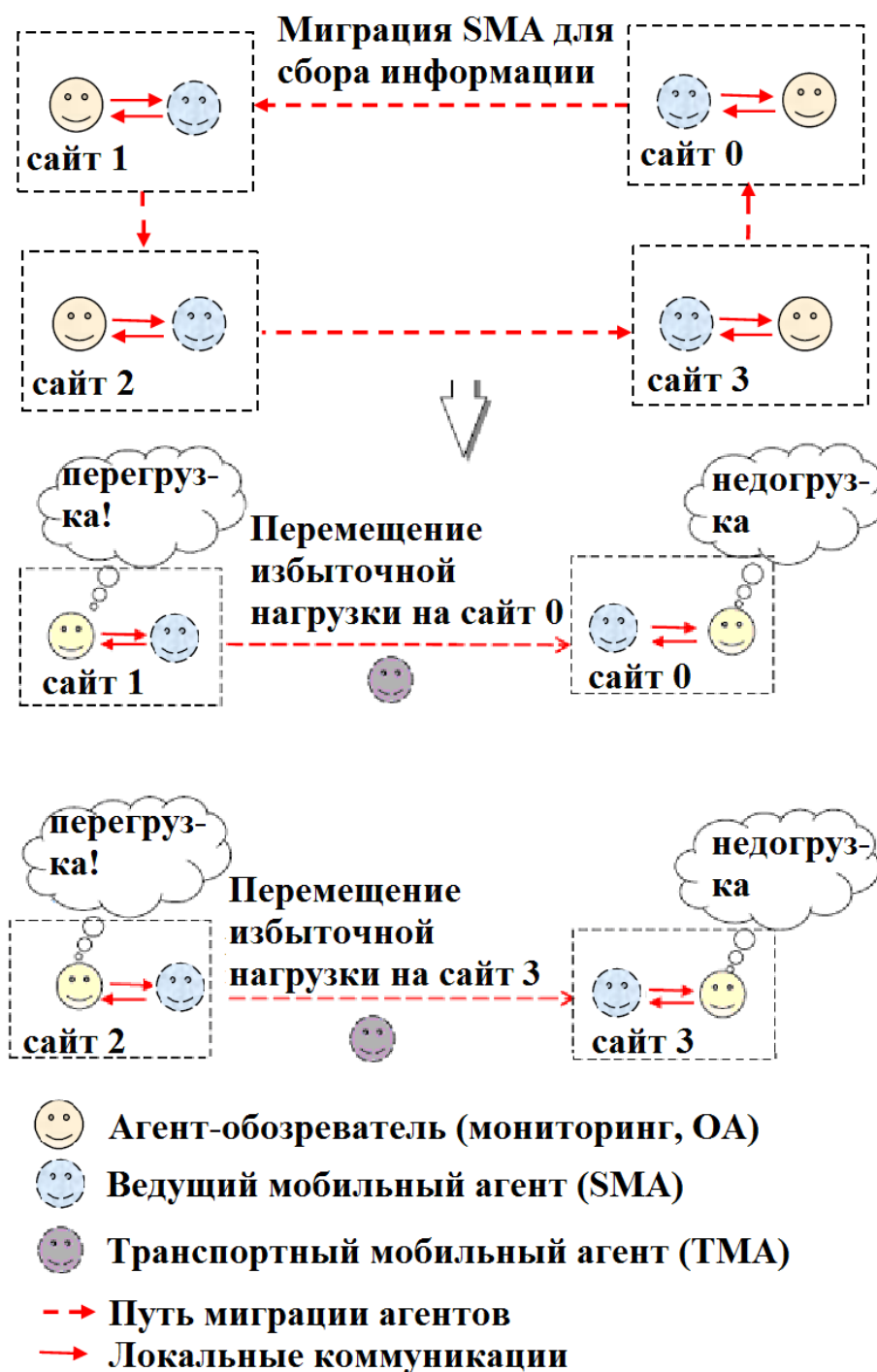
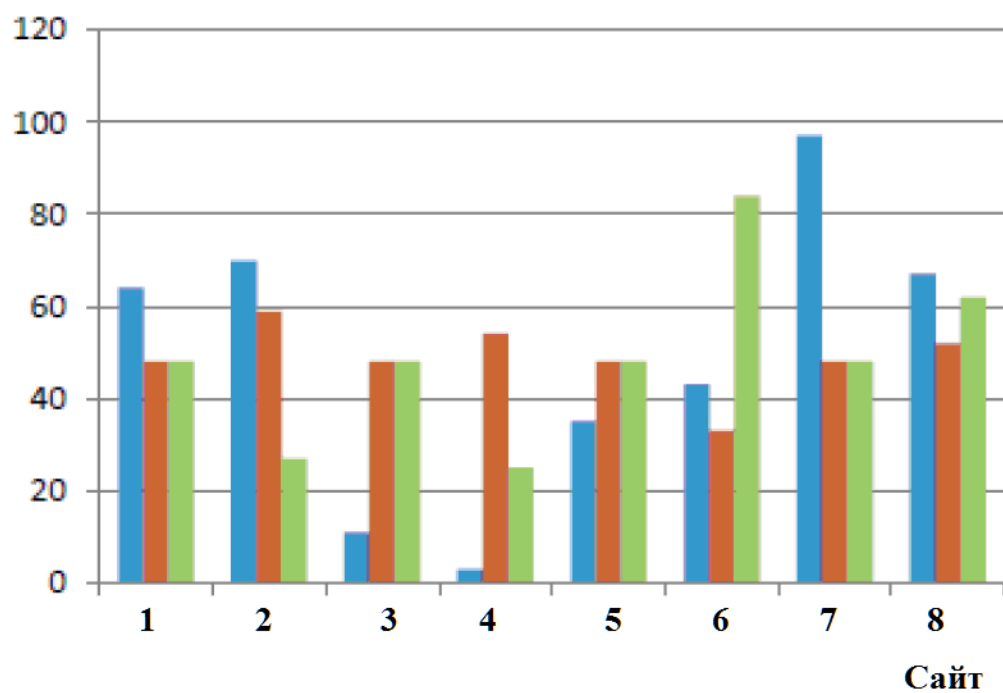
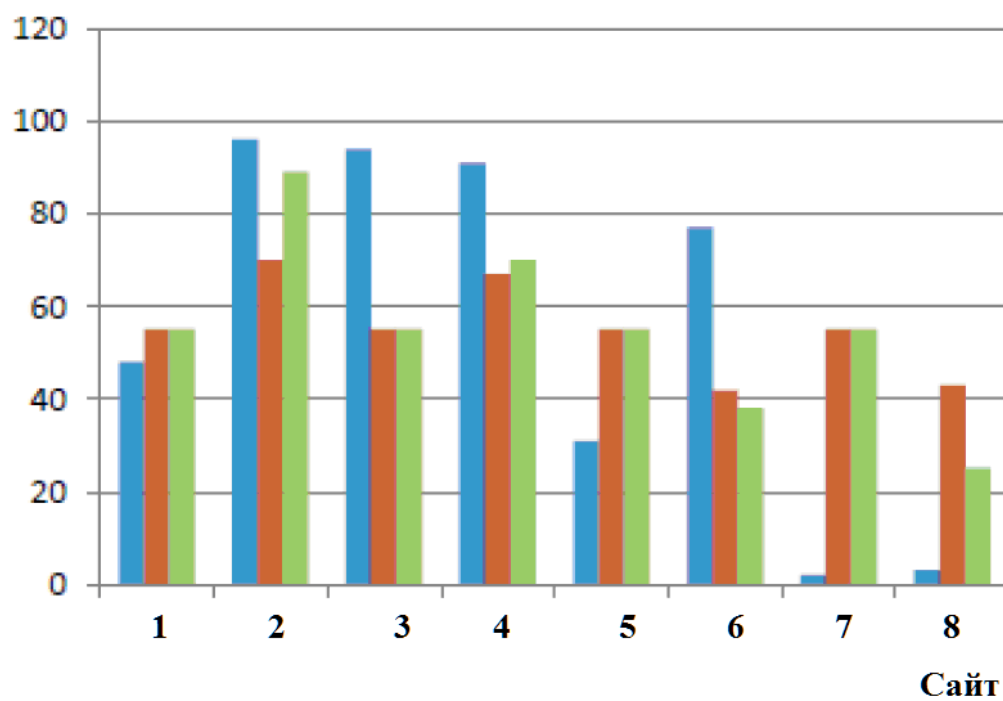


Рис. 3.2. Архитектура предложенной системы

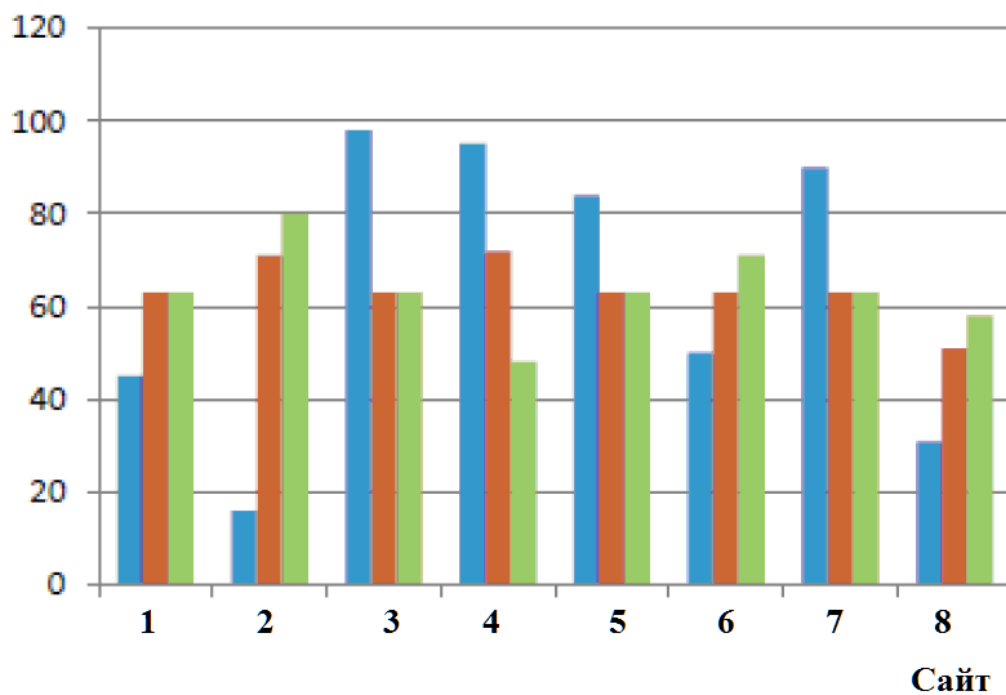
Коэффициент загрузки присваивается случайным образом каждому контейнеру. На рис. 3.3 представлено состояние нагрузок в три этапа: до применения LB и при применении предлагаемого LB с табличным подходом и без него. На третьем этапе (без таблицы) ТМА транспортирует перегрузку на первый найденный недогруженный участок.



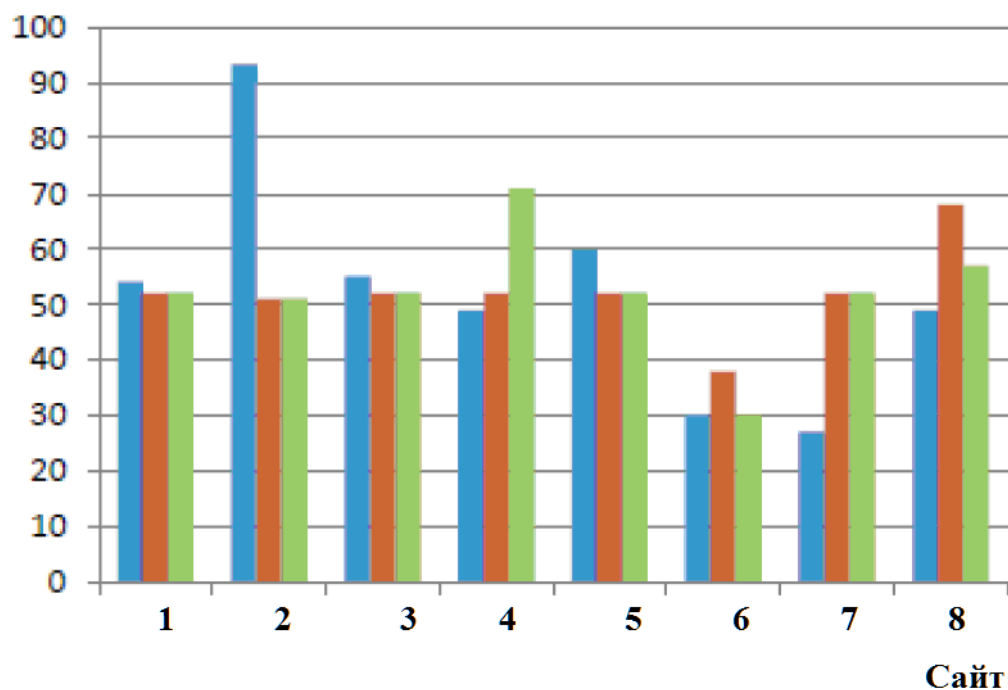
а) Состояние системы: исходное



б) Состояние системы: первый этап



в) Состояние системы: второй этап



г) Состояние системы: третий этап

Рис. 3.3. Состояние загрузки на трех этапах, после исходного: здесь и далее ■ - до балансировки; ■ - балансировка с таблицами; ■ - балансировка без таблиц

На рис. 3.3 показано, что на первом этапе нагрузки

несбалансированы (от очень высоких до очень низких нагрузок). На втором этапе нагрузки начинают балансироваться, что показывает влияние подхода LB. После этого, на третьем этапе, нагрузки более сбалансированы по сравнению с общим исходным состоянием системы, что демонстрирует эффективность предлагаемого табличного подхода при выборе правильного партнера для получения избыточной нагрузки.

3.2.2. Влияние способа балансировки на время отклика

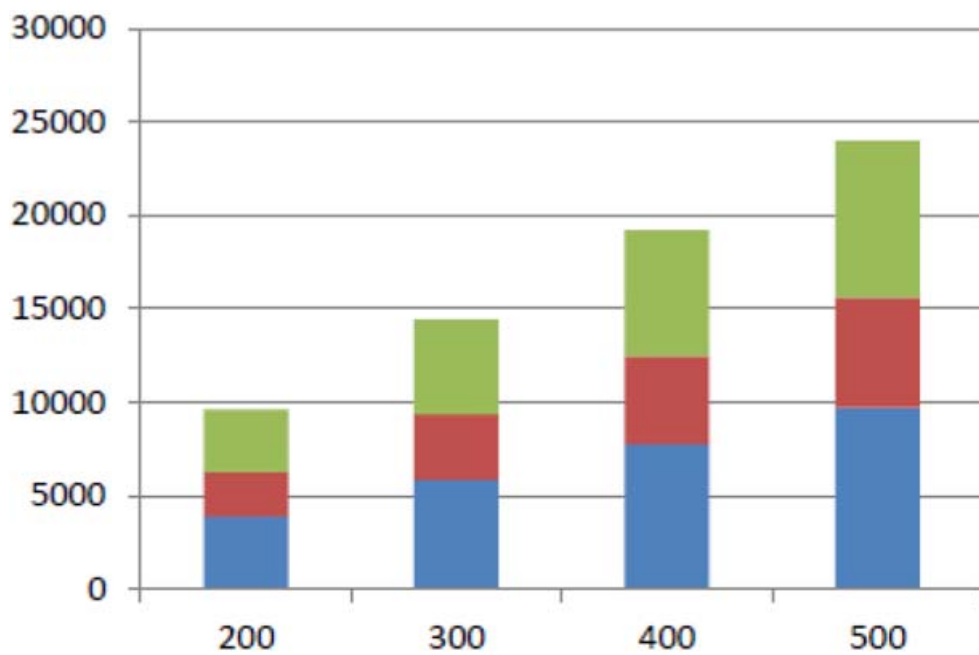
В распределенных вычислениях время отклика является существенной проблемой, и факт его сокращения является очень важным требованием в подходах к совершенствованию. Однако при больших нагрузках время отклика системы увеличивается. В среде такого типа набор задач распределяется между некоторыми узлами системы. Время отклика системы - это время, необходимое всем участвующим узлам для реализации этого набора задач.

Время отклика измеряем, используя следующую формулу:

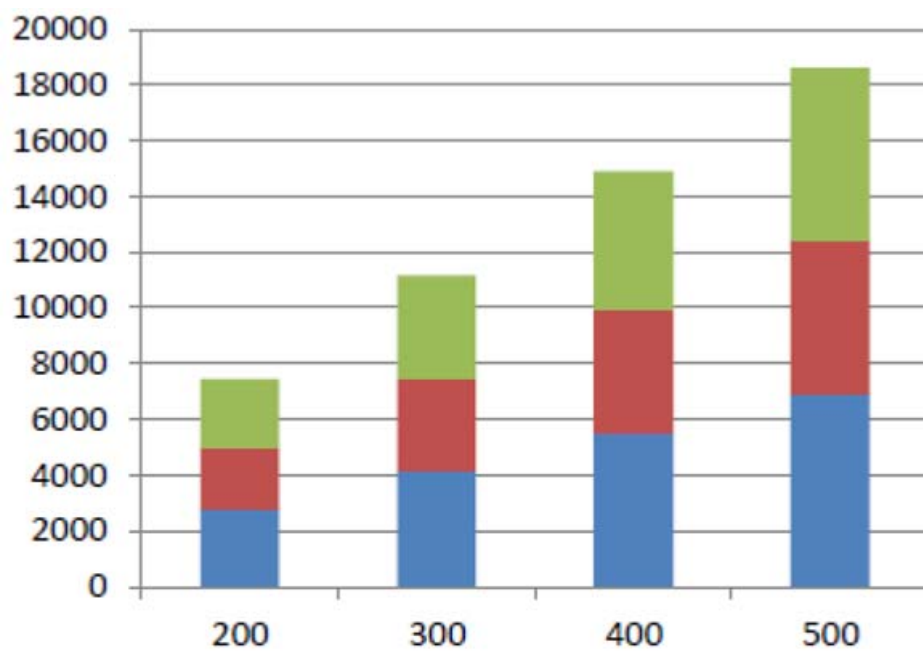
$$T = (NbTasks \times HighL \times ReqAT) / 100$$

- NbTask - общее количество задач,
- HighL - наибольший процент загрузки в этой системе,
- ReqAT - необходимое среднее время для выполнения одной задачи.

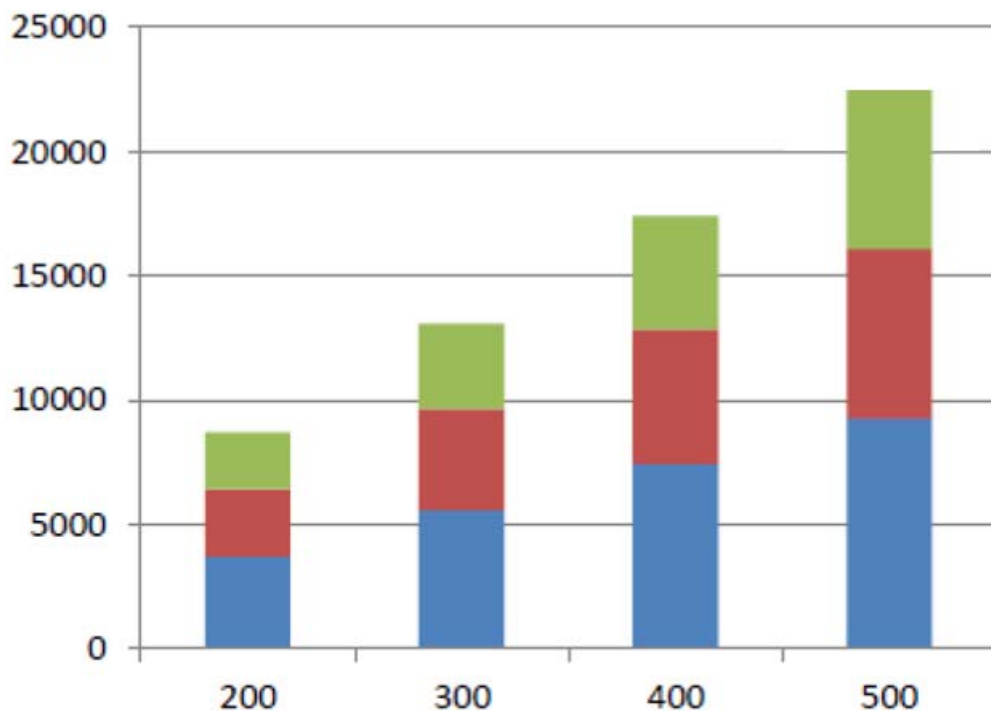
На рис. 3.4 показано изменение времени отклика на трех этапах при одновременном увеличении числа задач, которые должна выполнять эта система (между 200, 300, 400 и 500) для трех наборов случайных рабочих нагрузок. Результаты доказывают эффективность предложенного алгоритма LB в сокращении времени отклика по сравнению с двумя другими фазами.



а) Набор случайных рабочих нагрузок №1



б) Набор случайных рабочих нагрузок №2



в) Набор случайных рабочих нагрузок №3

Рис. 3.4. Время отклика как функция от числа заданий

3.2.3. Исследование предложенной методики

Основная цель применения процесса LB в распределенной системе заключается в том, чтобы все узлы работали с почти равными рабочими нагрузками. Другими словами, сводится к минимуму разница в долях рабочей нагрузки между узлами и, таким образом, исключается наличие узла с высокой рабочей нагрузкой, в то время как другой узел недостаточно загружен. В распределенной системе, где узлы работают совместно для выполнения определенного набора задач, балансировка рабочей нагрузки и, следовательно, загрузки между узлами имеет важное значение для сокращения времени выполнения (или времени отклика) задачи и, следовательно, всех запущенных задач. Поэтому в системе измеряется разница в показателях рабочей нагрузки между наиболее загруженными узлами и наименее загруженными.

В предлагаемой методике выбора перегруженный узел выбирает один из недогруженных узлов, куда передавать перегрузку (партнера)

Чтобы доказать эффективность предложенной методики, в этом подразделе она сравнивается с методом отбора, предложенным в [3.14] и в [3.21] с точки зрения различий в рабочей нагрузке и времени отклика.

Методика выбора [3.14] основана на сортировке недогруженных узлов в списке в соответствии с параметром привлекательности, а в [3.21] список сортируется в соответствии с коэффициентом использования. Затем каждый перегруженный узел использует этот список, чтобы найти наименее заряженный узел для передачи ему перегрузки. Когда агент миграции прибывает на выбранный узел, он проверяет, перегружен ли этот узел, чтобы принять полученную перегрузку или отклонить ее, если она недостаточно загружена. В случае отклонения агент миграции перемещается на следующий узел из отсортированного списка.

На рис. 3.5 представлена разница в показателях рабочей нагрузки между наиболее загруженным сайтом (узлом) и наименее загруженным сайтом в каждом моделировании (случае). Этот параметр сравнивается в три этапа: первый этап - перед применением LB, второй - при применении предлагаемого нами подхода и третий - при применении LB с использованием методов выбора [3.14, 3.21]. Уровень рабочей нагрузки присваивается каждому узлу случайным образом, и запущено восемь симуляций (8 случаев). Результаты на рис. 3.5 показывают, что различия в показателях рабочей нагрузки в большей степени уменьшаются после применения известного подхода, а затем значительно уменьшаются при применении предлагаемого подхода. Эти результаты доказывают эффективность предложенной методики выбора по сравнению с существующими методами выбора наиболее подходящего недогруженного узла для каждого перегруженного узла, и, таким образом, все узлы работают с почти одинаковой нагрузкой.

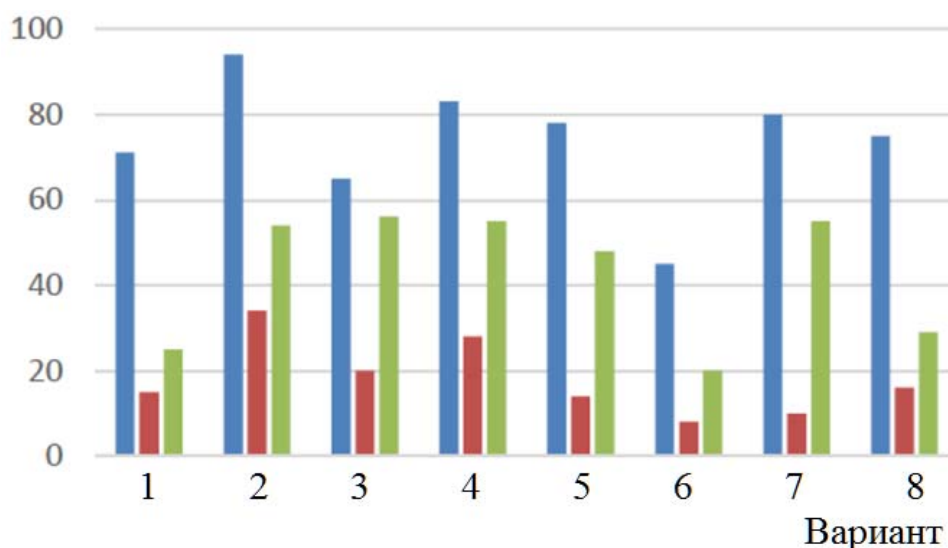
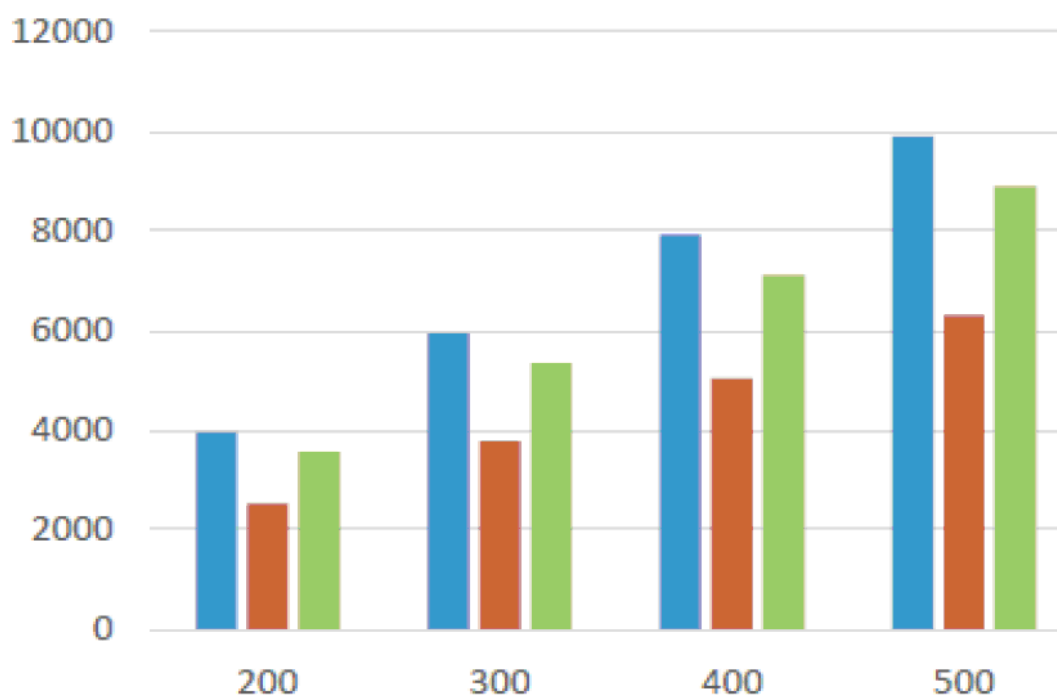
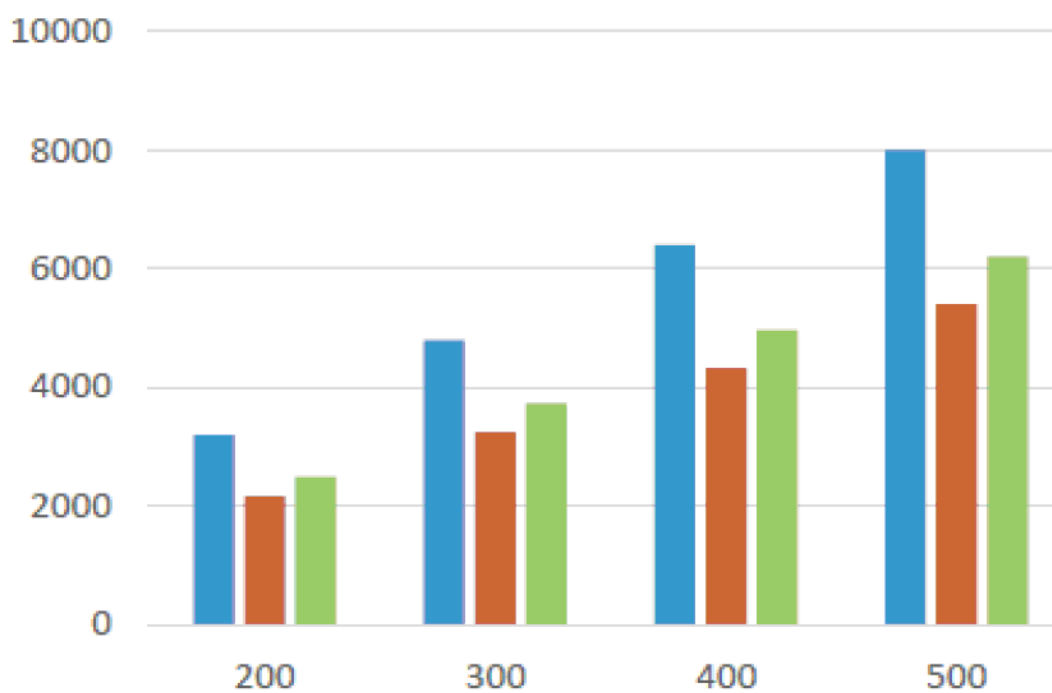


Рис. 3.5. Различия в показателях рабочей нагрузки между наиболее загруженными сайтами и наименее загруженными сайтами: здесь и далее ■ - до балансировки; ■ - наш подход; ■ - другие техники выбора

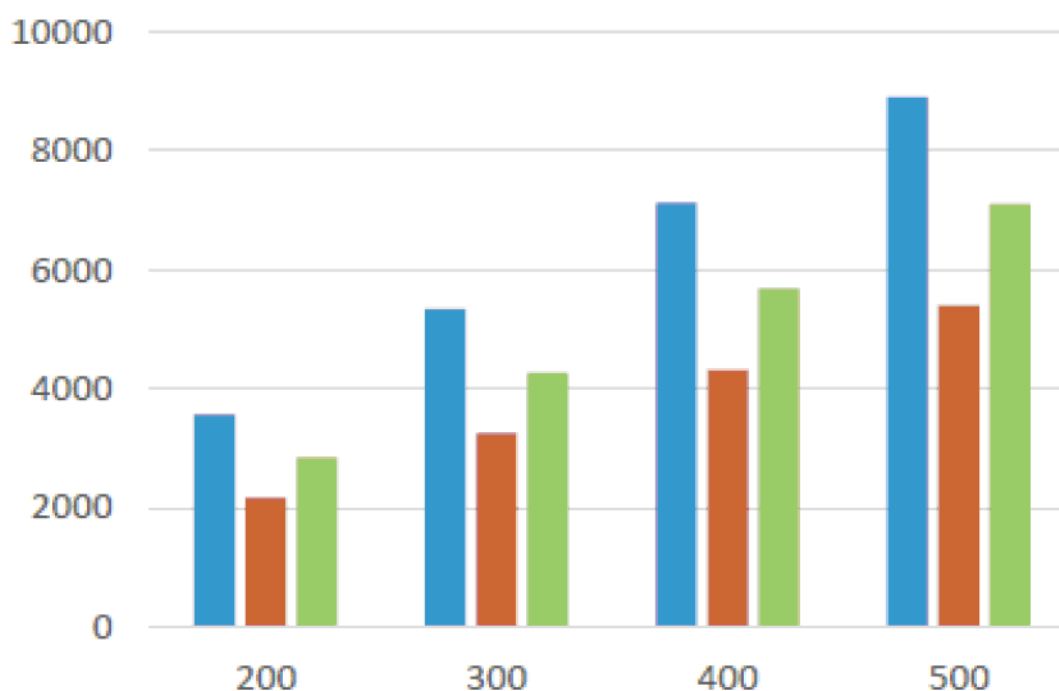
На рис. 3.6 показано влияние используемой методики выбора на время отклика при изменении количества заданий с 200, 300, 400 до 500. Далее объясняется значение времени отклика и определяется формула для его измерения. Результаты, представленные на рис. 3.6, соответствуют трем запущенным моделям с различными случайными рабочими нагрузками. На рис. 3.6 показано, что время отклика увеличивается с увеличением числа задач, требующих выполнения большего количества рабочих нагрузок. Кроме того, предлагаемый подход обеспечивает максимально сокращенное время отклика в трех запущенных симуляциях. Это доказывает эффективность предложенной методики выбора при выборе узла-партнера, и, таким образом, все узлы участвуют почти с одинаковыми усилиями в выполнении задачи, а следовательно выполнение набора задач происходит быстрее.



а) Набор случайных рабочих нагрузок №1



б) Набор случайных рабочих нагрузок №2



в) Набор случайных рабочих нагрузок №3

Рис. 3.6. Влияние метода выбора на время отклика в зависимости от количества задач

3.3. Миграция услуг на основе мобильных агентов для обеспечения устойчивости к внешним и внутренним возмущениям

Рассмотрена проблема миграции услуг на основе мобильных агентов. В системах, которые оказывают услуги в реальном масштабе времени, где показатель QoS крайне важен для пользователя, возникают ситуации перегрузки платформ, на которых функционирует информационное и программное обеспечение, оказывающее услуги. Поддержание QoS на требуемом уровне требует перемещения услуг между платформами, от перегруженных к недогруженным. Для эффективного решения задачи перемещения необходим набор методов и инструментов. Целью работы является создание методологии перемещения в реальном масштабе времени с поддержанием QoS на требуемом уровне. Раздел

содержит подход к перемещению с использованием мобильных агентов. Важно, что схема перемещения формируется не центром управления, а коллективное создание схемы перемещения посредством мобильных агентов. Такая коллективная работа порождает план перемещения, учитывая параметры надежности, безопасности и загруженности платформ. Важным условием является то, что при перемещении служб не произойдет никаких изменений у остальных агентов. Фактически такое перемещение агентов сохраняет не только параметры QoS, но и обеспечивает повышение отказоустойчивости всей системы. Таким образом, раздел содержит подход к перемещению с использованием мобильных агентов в реальном масштабе времени с поддержанием QoS на требуемом уровне, обеспечивая повышение отказоустойчивости всей системы.

Настоящее исследование показывает применение миграции мобильных агентов для обеспечения доступности услуг. Таким образом, работа сосредоточена на том, как мобильность агентов помогает динамически перемещать службы с некорректных платформ на другие корректные платформы, чтобы повысить доступность услуг. Она дополняет современные исследования в области использования мобильных агентов для повышения отказоустойчивости системы.

3.3.1. Архитектура системы мобильных агентов

На рис. 3.7 показаны основные компоненты архитектуры, в которой СОС – сервер управления событиями, СУР – сервер управления ресурсами, МА – мобильный агент, СИА – среда исполнения агента, П – платформа, ОВОУ - обнаружение вторжений и оценка ущерба, МС – мониторинг системы.

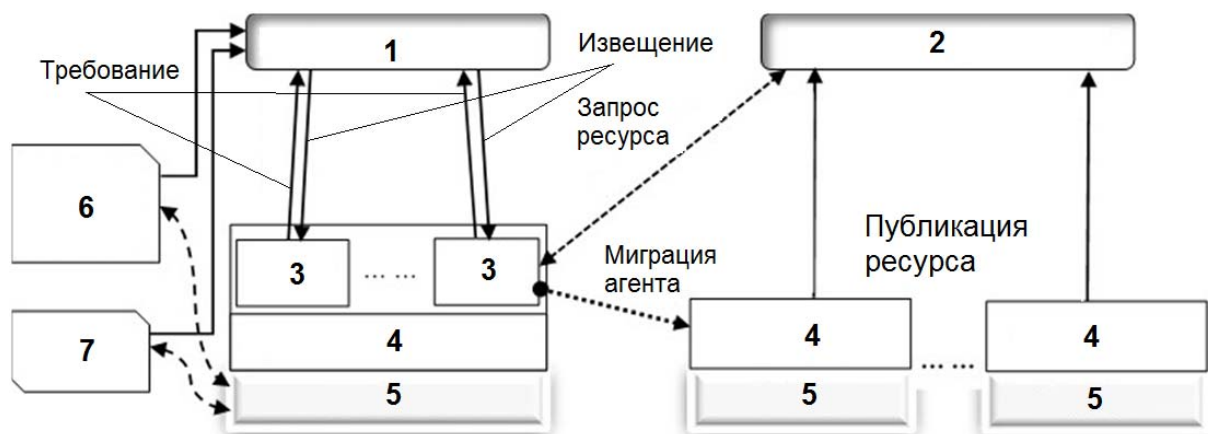


Рис. 3.7. Архитектура системы мобильных агентов: 1 – СОС; 2 – СУР; 3 – МА; 4 - СИА; 5 – П; 6 - ОВОУ; 7 – МС

Местоположение агента – в автономном контейнере на платформах, но в каждой временной точке – на единственной платформе. RMS и EMS сервера также решают задачи обслуживания и помогают агентам в межплатформенных перемещениях. EMS начинает процесс перемещения, RMS снабжает агентов ресурсами в рамках контейнеров.

Модель запроса на публикацию используется для управления ресурсами службой управления правами. В то время как каждая платформа публикует набор ресурсов (наряду с качеством их обслуживания), которые она может предложить через службу управления правами, агент запрашивает ресурсы и решает, может ли платформа поддерживать свои операции. В отличие от модели push-уведомлений о регистрации EMS, запрос через службу управления правами предназначен для получения агентом информации о ресурсах, предоставляемых платформой. В случае, если на текущей платформе агента было обнаружено вторжение в систему безопасности или платформа не может обеспечить агенту удовлетворительный уровень сервиса, агент должен запросить службы управления правами, чтобы определить возможные платформы для миграции.

3.3.2. Миграция услуг на основе мобильных агентов

В этом разделе обсуждаются детали схемы миграции услуг на основе мобильных агентов. Миграция мобильного агента начинается с уведомления EMS о серьезном повреждении платформы, на которой в настоящее время работает агент, или обнаружении того, что текущая платформа не может предоставлять желаемые услуги. Затем агент должен решить, на какие возможные платформы следует перейти. Учитывая взаимодействие и функциональную зависимость между несколькими агентами, работающими над одной и той же задачей или сотрудничающими друг с другом, группе агентов необходимо составить план совместной миграции. В следующих разделах обсуждаются принятие решений о миграции мобильных агентов и диспетчеризацию агентов.

Принятие решений о миграции локальных и совместных агентов

Решение агента о переходе на другую, более безопасную и надежную платформу основано на двух важных факторах.

Первый из них - это ощущение агентом необходимости миграции. Если текущая платформа была серьезно повреждена и атака может привести к разрушению других ресурсов на платформе, агент должен перейти на другую платформу. Таким образом, агент может минимизировать любые потенциальные потери и постоянно предоставлять пользователям высококачественные услуги.

Еще одним важным фактором для принятия решения о переносе агента является доступность платформ, которые могут предоставить необходимые агенту ресурсы. Агент ищет и идентифицирует те платформы (через службу управления правами), которые удовлетворяют его функциональным требованиям и требованиям к качеству. Затем он решает, на какие платформы он может перейти. Это называется локальным

решением о миграции, поскольку решение принимается одним агентом на основе его локальной информации. Тем не менее, ресурсы, требуемые агентом, должны указывать типы ресурсов и соответствующие требования к качеству. Результатом решения о локальной миграции является список упорядоченных платформ (с точки зрения желаемых предпочтений), на которые агент может перейти.

После того, как каждый отдельный агент примет решение о локальной миграции, группа агентов примет совместное решение о миграции в отношении окончательного плана миграции. Групповое решение необходимо из-за взаимозависимости, сотрудничества и даже отношений исключения между одноранговыми агентами в группе для предоставления пользовательских услуг. Например, один агент может функционально зависеть от других агентов (отношение "зависит от").

Группе агентов может потребоваться работать на одной платформе (связь атомарности) по таким причинам, как совместное использование общих ресурсов, тесное взаимодействие друг с другом, выполнение набора групповых действий или выполнение определенных общих операций по обеспечению безопасности и конфиденциальности. Некоторые агенты даже имеют отношения взаимного исключения по таким причинам, как разделение обязанностей, функциональная и социальная изоляция или просто организационная политика; поэтому они не должны мигрировать на одну и ту же платформу. В предложенной схеме принятия решений эти и другие виды операционных отношений агентов представлены в виде набора ограничений, которые не должны нарушаться окончательным планом миграции для группы агентов. Совместное принятие решений о миграции заключается в определении такого окончательного плана миграции для каждого агента в группе.

Решение о перемещениях может быть совершенно децентрализованным, с участием заинтересованных агентов. Вместе с тем,

поскольку система принятия решений работает как система реального времени, в случае возникновения недетерминированных состояний системы принятие решений может быть проведено не в темпе процесса, а медленнее, и тогда совершенная миграция понизит уровень качества обслуживания. Таким образом, целесообразно использовать гибридно-централизованный механизм принятия решений с их объединением и интеграцией силами специального агента-координатора. На локальном уровне (без вмешательства специального агента-координатора) решение о локальном перемещении принимается только локальными агентами. И только после этого выделяется специальный агент-координатор, который принимает от всех локальных мобильных агентов группы и реестр платформ (рис. 3.8).

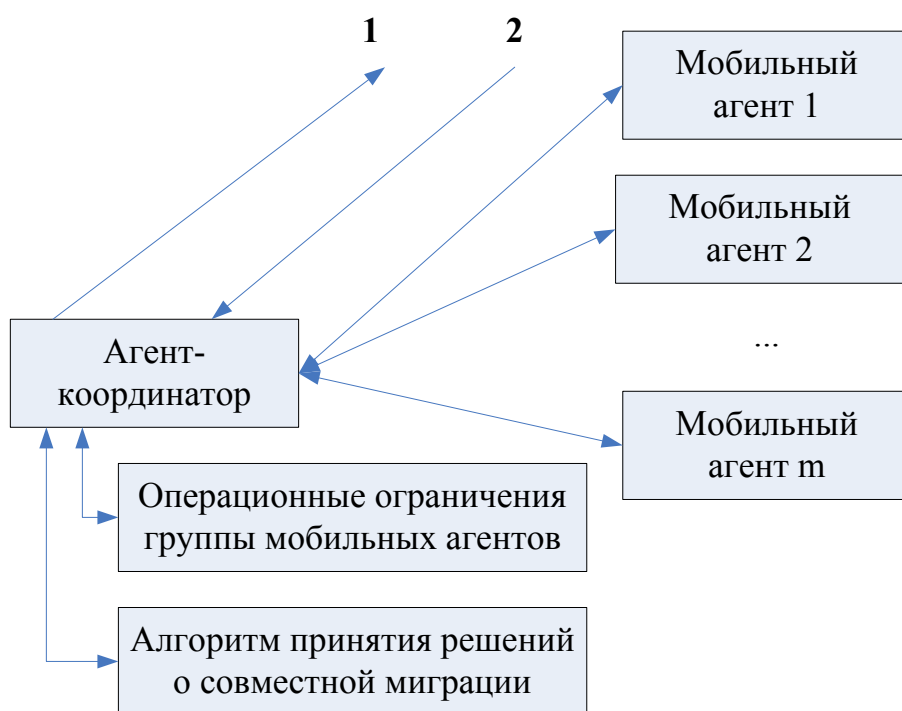


Рис. 3.8. Принятие решений о совместной миграции мобильного агента с агентом-координатором: 1 – локальные сведения, требования и ограничения миграции (ID агента, множество платформ для миграции); 2 – конечный план миграции

Далее специальный агент-координатор учитывает известные ограничения мобильных агентов и пробует создать способ групповой миграции, подходящий для всей группы и входящих в нее мобильных агентов.

Разработан алгоритм для совместного принятия решений о миграции, выполняемый агентом-координатором. Алгоритм принимает в качестве входных данных решения о локальной миграции $\{S_1, \dots, S_i, \dots, S_m\}$ от всех m агентов в группе и набор ограничений R для группы агентов. Результатом является окончательный план миграции агентов, если таковой существует, или сообщение, указывающее, что такой план невозможен, учитывая решения о локальной миграции текущих агентов и ограничения на работу группы. В R указаны три типа ограничений [3.29]: (1) правило атомарности – набор агентов должен быть перенесен на одну и ту же платформу, (2) правило зависимости – агент функционально зависит по крайней мере от одного из нескольких агентов; следовательно, он должен мигрировать на ту же платформу, что и агент, от которого он зависит, и (3) правило исключения – два или более агентов не должны быть перенесены на одну и ту же платформу. По сути, алгоритм принятия решения о совместной миграции рекурсивно выбирает одну платформу из каждого списка возможных платформ, предоставляемых одним агентом, называемого предварительным планом миграции, а затем проверяет этот предварительный план по каждому правилу в списке ограничений. Если какое-либо ограничение нарушено, этот предварительный план неосуществим, и алгоритм переходит к следующему предварительному плану миграции до тех пор, пока не будут исчерпаны все возможные предварительные планы или не будет определен окончательный возможный план миграции. В последнем случае окончательный план миграции отправляется каждому агенту, чтобы он мог начать миграцию.

Алгоритм: Совместное принятие решений о миграции

Вход: (1) Решение о локальной миграции S_i ($i=1\dots m$) от каждого агента A_i ($i=1\dots m$), где S_i содержит набор платформ, на которые A_i может мигрировать, а m представляет количество мобильных агентов в группе; (2) ограничения для работы группы агентов R ;

Выход: План миграции для каждого агента A_i для перехода на платформу P_j ($j=1\dots n$), где n представляет количество доступных платформ.

Начало

1. Определить предварительный план миграции WP с одной платформой от каждого S_i . Первоначально $WP = \emptyset$;

2. Для каждого набора платформы S_i , отправленного агентом A_i ($i=1\dots m$), выполнить:

2.1. Выбрать платформу $P'_i \in S_i$ и добавить P'_i к WP , т.е.

$$WP = WP \cup \{P'_i\};$$

3. Для каждого ограничения $r \in R$ выполнить:

3.1. Если r является ограничением зависимости, т.е. агент ag зависит по крайней мере от одного агента в наборе агентов A' , выполнить:

3.1.1. Если $\forall ag' \in A'$, соответствующие платформы ag и ag' в WP различны, выполнить:

$$\text{Reset } WP = \emptyset;$$

Возврат к шагу 2 для поиска следующего предварительного плана миграции;

3.2. Если r является атомарным ограничением, т.е. все агенты должны быть перенесены на одну и ту же платформу, выполнить:

3.2.1. Если для некоторой пары платформ $P, P' \in WP$ и $P \neq P'$, выполнить

$$\text{Очистить } WP = \emptyset;$$

Возврат к шагу 2 для поиска следующего предварительного плана миграции;

3.3. Если r представляет ограничение исключения, т.е. два или более агента не могут быть перенесены на одну и ту же платформу, выполнить:

3.3.1. Если для некоторой пары платформ $P, P' \in WP$ и $P = P'$,

Очистить $WP = \emptyset$;

Возврат к шагу 2 для поиска следующего предварительного плана миграции;

4. Если предварительный план миграции WP удовлетворяет всем правилам ограничений в R , выполнить:

4.1. Для каждого элемента $P' \in WP, P' \in S_i$, выполнить:

4.1.1. Уведомить агента A_i о платформе P_i для миграции и конкретном расписании миграции;

5. Иначе, выполнить:

5.1. Если доступно больше ресурсов, выполнить:

Обновить каждый S_i новыми ресурсами в системе, которые может использовать агент A_i ;

Вернуться к шагу 1 для следующего раунда принятия решений о миграции с новыми выделенными ресурсами

Окончание

В случае, если в первом раунде алгоритма совместного планирования миграции невозможно определить возможный план миграции, группа агентов может перезапустить процесс, если доступно больше системных ресурсов (шаг 5 алгоритма). Агенты проверяют в службе управления правами, доступны ли некоторые новые платформы (некоторые серверы резервного копирования, возможно, были запущены в рамках плана реагирования на системные инциденты) или доступны некоторые новые ресурсы (например, доступны дополнительные ресурсы

системных компонентов или служб). Этот процесс может повторяться несколько раз до тех пор, пока не будет составлен план совместной миграции или не будет достигнута альтернативная договоренность о перемещении некоторых агентов с поврежденных платформ. В определенных ситуациях мобильным агентам может потребоваться проявить гибкость, перенастроив некоторые функции и, таким образом, ослабив определенные правила ограничений. В качестве альтернативы, услуги пользователя могут временно предлагаться на минимальном, но приемлемом уровне. В этих случаях может быть возможен совместный план миграции с учетом добавления ресурсов или смягченных правил ограничений.

Диспетчеризация агентов

После составления окончательного плана миграции агентов каждый мобильный агент получает уведомление о платформе, на которую он переходит, а также о конкретном расписании миграции, например о порядке миграции. Например, если агент A1 функционально зависит от агента A2, то A2 необходимо сначала перенести на целевую платформу до A1, чтобы A2 мог сначала восстановить свои операции на целевой платформе. Если несколько агентов мигрируют с одной и той же исходной платформы, необходима упорядоченная отправка. С другой стороны, если агенты с нескольких платформ перемещаются на одну и ту же платформу, их следует развертывать в определенном порядке, который учитывает их срочность и функциональную зависимость от других агентов на той же платформе.

Технически отправка агента на новую платформу включает в себя отключение текущих запросов пользователей на обслуживание агента и перенаправление их на новую платформу, сохранение состояний агента, упаковку программ агента и передачу кода агента на новую платформу. В

зависимости от хранилища данных, схеме миграции агента может потребоваться также переместить пространство данных агента на новую платформу.

Однако, если данные агента хранятся удаленно, а доступ к данным осуществляется через сетевое подключение (например, хранилище данных в облаке), ему необходимо только сохранить текущий статус доступа к данным, а затем агент непрерывно получает доступ к данным с новой платформы. В последнем случае, поскольку нет необходимости физически перемещать данные, это значительно снижает нагрузку на сеть для миграции агентов и, следовательно, ускоряет весь процесс миграции мобильных агентов.

3.3.3. Система проверки концепции для моделирования предлагаемой схемы миграции агентов

Разработана система проверки концепции для моделирования предлагаемой схемы миграции агентов. В моделировании каждый агент представляет собой фрагмент исполняемого кода, который живет в определенной среде, то есть на платформе мобильного агента. У агента есть срок службы, то есть он может быть создан, развернут, выполнен и, наконец, уничтожен. Использована среда Aglets [3.30] в качестве платформы мобильных агентов, которая предоставляет агентам набор ресурсов и контролирует жизненный цикл агентов. Aglets - это агентная платформа и библиотека на основе Java для создания мобильных приложений на основе агентов [3.31]. Aglet - это агент Java, который специально разработан для поддержки мобильности, т.е. позволяет агенту перемещаться по хостам одной или нескольких сетей.

Для простоты EMS и RMS интегрированы в один агент. Программы могут создавать произвольное количество мобильных агентов, выполняющихся в одном или нескольких контекстах Aglets на платформе.

Чтобы запустить миграцию, EMS отправляет предупреждение каждому агенту, работающему на платформе, которая предположительно повреждена в результате инцидента безопасности. Агент принимает решение о локальной миграции на основе доступных платформ и ресурсов, которые они предоставляют. Внедрена информация о ресурсах платформы в файл, управляемый службой управления правами и доступный мобильным агентам. Структура прототипа системы управления миграцией – системы моделирования – приведена на рис. 3.9.

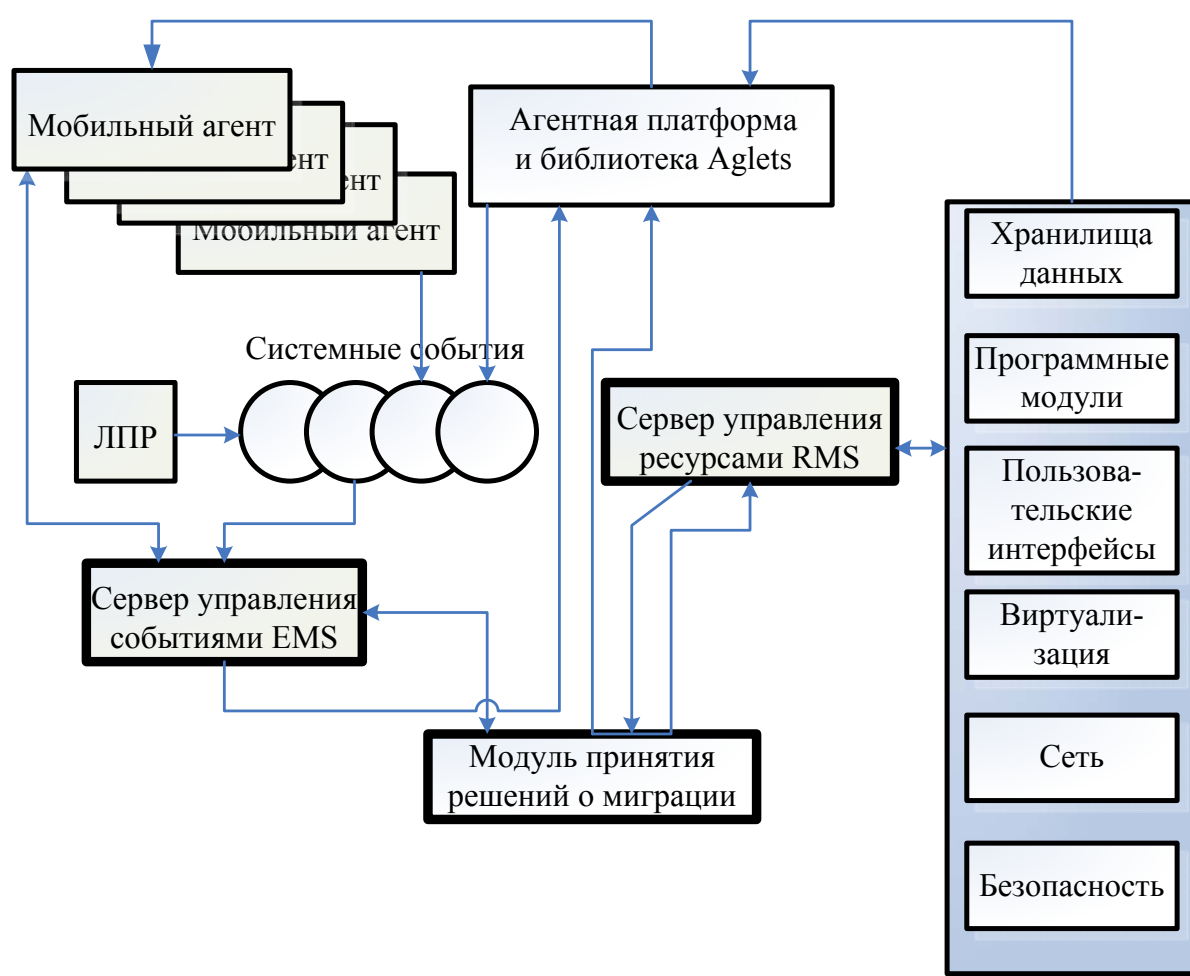


Рис. 3.9. Структура прототипа системы управления миграцией – системы моделирования

Чтобы убедиться, что предлагаемая схема миграции агентов может быть применена к общим случаям с различными ресурсами, требуемыми

агентами, и ресурсами, предоставляемыми платформами, программы случайным образом генерируют типы ресурсов, а также количество и качество каждого из них. Поскольку информация о ресурсах хранится в файле ресурсов, агенту необходимо получить доступ к файлу только для того, чтобы принять решение о локальной миграции в отношении набора платформ, на которые он может перейти. Aglets предоставляет механизм обмена сообщениями для обмена данными между агентами. Таким образом, этот механизм может быть использован для каждого агента для отправки этого набора платформ агенту-координатору. Также использован встроенный метод обработки сообщений Aglets для каждого агента для обработки полученных сообщений. Этот метод по существу определяет действия, предпринимаемые каждым агентом.

Диспетчеризация мобильных агентов обеспечивается API-интерфейсами Aglets. Сложные и подробные операции перемещения агентов (сериализация объектов, синхронизация и перемещение данных) выполняются в среде исполнения Aglets. В качестве примера моделирования первоначально 11 агентов (включая EMS и RMS на одном агенте) работали на платформе (как показано в верхней части рис. 3.10). Для поддержки миграции агентов в системе были доступны 4 новые платформы. После запуска наших программ моделирования 10 мобильных агентов были отправлены на соответствующие им новые платформы, из которых 8 агентов переехали на одну платформу (как показано в середине рис. 3.10), а 2 других агента переехали на другую платформу (как показано в нижней части рис. 3.10). Aglets также обеспечивает операцию возврата, которая позволяет отправленным агентам вернуться на свои исходные платформы. Перемещение полезно для балансировки нагрузки системы (для восстановления служб агентов до их первоначального распределения рабочей нагрузки) и оптимизации обслуживания (чтобы убедиться, что пользовательские программы находятся ближе к агентам или чтобы

убедиться, что программы агентов находятся ближе к данным, которые они используют). Хотя система проверки концепции не предназначена для использования в производственном приложении, она демонстрирует возможность и эффективность перемещения агентов с одной платформы на другую с помощью платформы мобильных агентов, такой как Aglets.

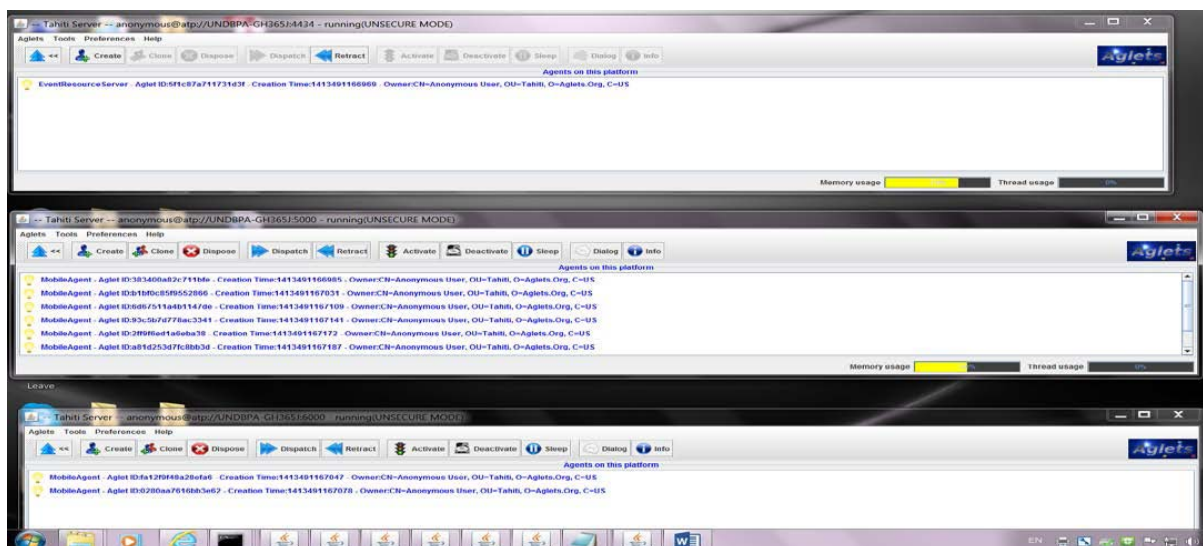


Рис. 3.10. Общий вид моделирования системы управления 10 мобильными агентами, перенесенными на 2 платформы на главном компьютере

3.4. Выводы

Управление нагрузкой является одним из ключевых факторов повышения производительности системы. Его основными целями являются увеличение времени выполнения задач и максимальное использование системных ресурсов.

В работе исследовано использование технологии мобильных агентов в области балансировки нагрузки. Одной из важных мотиваций этой парадигмы является минимизация удаленной связи, что позволяет экономить потребление полосы пропускания, что благоприятно для

эффективной системы балансировки нагрузки. По этой причине в предлагаемое решение интегрирован мобильный агент, роль которого заключается в сборе информации о нагрузках для построения глобального видения нагрузки на систему, что снижает затраты на связь по сравнению с классическим методом сбора. Кроме того, стационарные агенты должны знать о глобальной нагрузке, чтобы они могли определять состояние баланса машины и, таким образом, выбирать задачи для переноса. В предложенном решении мобильный наблюдательный агент информирует только перегруженные хосты, что приводит к сокращению трафика по сравнению с теми подходами, которые информируют все хосты системы.

Другими преимуществами мобильных агентов являются надежность и отказоустойчивость, которые необходимы в системе балансировки нагрузки, чтобы она могла продолжать работать, когда один из участников отключен. Мобильные агенты также обладают преимуществом масштабируемости, они хорошо адаптируются как к небольшим сетям, так и к крупномасштабным сетям. Эти преимущества позволяют использовать мобильных агентов, чтобы иметь расширяемую и надежную систему балансировки нагрузки.

В крупномасштабной сети увеличение числа контролирующих агентов позволяет сократить время миграции агентов и избежать увеличения размера кода агента. Это позволяет улучшить балансировку нагрузки. Сравнительная оценка результатов показала эффективность подхода к балансировке нагрузки. Кроме того, показано его влияние на сокращение задержки времени выполнения, что является важным фактором в распределенных системах. Кроме того, продемонстрировано влияние предлагаемых методов отбора по сравнению с существующими методами отбора на балансировку рабочих нагрузок и сокращение времени отклика системы.

Рассмотрена проблема миграции услуг на основе мобильных

агентов. В системах, которые оказывают услуги в реальном масштабе времени, где показатель QoS крайне важен для пользователя, возникают ситуации перегрузки платформ, на которых функционирует информационное и программное обеспечение, оказывающее услуги. Поддержание QoS на требуемом уровне требует перемещения услуг между платформами, от перегруженных к недогруженным. Для эффективного решения задачи перемещения необходим набор методов и инструментов. Целью работы является создание методологии перемещения в реальном масштабе времени с поддержанием QoS на требуемом уровне. Статья содержит подход к перемещению с использованием мобильных агентов. Важно, что схема перемещения формируется не центром управления, а коллективное создание схемы перемещения посредством мобильных агентов. Такая коллективная работа порождает план перемещения, учитывая параметры надежности, безопасности и загруженности платформ. Важным условием является то, что при перемещении служб не произойдет никаких изменений у остальных агентов. Фактически такое перемещение агентов сохраняет не только параметры QoS, но и обеспечивает повышение отказоустойчивости всей системы. Таким образом, представлен подход к перемещению с использованием мобильных агентов в реальном масштабе времени с поддержанием QoS на требуемом уровне, обеспечивая повышение отказоустойчивости всей системы.

Разработан механизм перемещения мобильных агентов для сохранения качества обслуживания или исключения нарушения работоспособности платформ в случае штатных или нерегламентированных возмущений (системные сбои, внешние атаки, например). В таких ситуациях мобильные агенты будут мигрировать с полностью или частично утративших работоспособность платформ на корректно функционирующие платформы для поддержания QoS на

требуемом уровне.

Разработана архитектура системы, определены процедуры миграции агентов и проведено моделирование для предлагаемой схемы миграции служб на основе агентов.

Список источников к главе 3

3.1. Van Steen M, Tanenbaum AS (2016) A brief introduction to distributed systems. Computing. vol. 98, no. 10, pp. 967-1009.

3.2. Anjomshoa MF, Salleh M, Kermani MP (2015) A Taxonomy and Survey of Distributed Computing Systems. Journal of Applied Sciences, Vol. 15, pp. 46-57.

3.3. Samolej S. and Rak T. (2009) Simulation and Performance Analysis of Distributed Internet Systems Using TCPNs, Informatica, vol. 33, pp. 405-415.

3.4. Hakansson A, Hartung R (2013) Agent and Multi-Agent Systems in Distributed Systems - Digital Economy and E-Commerce. Publisher: Springer-Verlag Berlin Heidelberg.

3.5. Hajduk M, Sukop M, Haun M (2019) Cognitive Multi-agent Systems : Structures, Strategies and Applications to Mobile Robotics and Robosoccer. In Series of Studies in Systems, Decision and Control.

3.6. Wided A. and Okba K. (2019) A novel Agent Based Load Balancing Model for maximizing resource utilization in Grid Computing, Informatica, vol. 43, no. 3.

3.7. M. A. Salehi, H. Deldari, and B. M. Dorri (2009) Balancing load in a computational grid applying adaptive, intelligent colonies of ants, Informatica, vol. 33, no. 2, pp. 159–168.

3.8. Aghdashi A, Mirtaheri SL (2019) A Survey on Load Balancing in Cloud Systems for Big Data Applications. In: Grandinetti L., Mirtaheri S., Shahbazian R. (eds) High-Performance Computing and Big Data Analysis. TopHPC 2019. Communications in Computer and Information Science, vol 891. Springer, Cham

3.9. Couturier R, Giersch A, Hakem M (2018) Best effort strategy and virtual load for asynchronous iterative load balancing. Journal of Computational Science. Vol. 26, Pages 118-127.

3.10. Priya V, Kumar CS, Kannan R (2019) Resource scheduling algorithm with load balancing for cloud service provisioning. *Appl Soft Comput* Vol. 76, pp. 416–424.

3.11. Ramos AG, Silva E, Oliveira JF (2018) A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, Vol. 266(3), pp. 1140–1152.

3.12. Hsieh H, Chiang M (2019) The Incremental Load Balance Cloud Algorithm by Using Dynamic Data Deployment. *J Grid Computing* Vol. 17, pp. 553–575.

3.13. Semchedine F, Bouallouche-Medjkoune L, Tamert M, Mahfoud F, Aïssani D (2015) Load balancing mechanism for data-centric routing in wireless sensor networks. *J. Comput. Electrical Eng.* Vol. 41, pp. 395–406.

3.14. Shen X-J et al (2014) Achieving dynamic load balancing through mobile agents in small world P2P networks, *Comput. Netw.*, vol. 75, pp. 134-148, Dec.

3.15. Ali M, Bagchi S (2019) Design and analysis of distributed load management: Mobile agent. based probabilistic model and fuzzy integrated model. *Appl Intell*, Vol. 49, pp. 3464-3489.

3.16. Metawei MA, Ghoneim SA, Haggag SM, Nassar SM (2012) Load balancing in distributed multi-agent computing systems, *Ain Shams Engineering Journal*, Vol. 3, pp. 237-249, May 2012.

3.17. Youssfi M, Bouattane O, Bensalah, M (2015) Efficient Load Balancing Algorithm for Distributed Systems Using Mobile Agents , *Advanced Studies in Theoretical Physics* Vol. 9, no. 5, pp.245 – 253.

3.18. Younes H, Bouattane O, Youssfi M, Illoussamen E (2017) New load balancing framework based on mobile AGENT and ant-colony optimization technique. *Intelligent Systems and Computer Vision (ISCV)*, Fez, 2017, pp. 1-6.

3.19. Afriansyah MF, Somantri M, Riyadi MA (2017) Model of load balancing using reliable algorithm with multi-agent system. *IOP Conference*

Series: Materials Science and Engineering, Volume 190, conference 1.

3.20. Nehra N, Patel RB, Bhat VK (2008) Load Balancing in Heterogeneous P2P Systems using Mobile Agents. *International Journal of Electrical and Computer Engineering* Vol:2, No:8, pp. 2740-2745.

3.21. Li H (2011) Load Balancing Algorithm for Heterogeneous P2P Systems Based on Mobile Agent. in *IJCA Proceedings on International Conference on Electronics, Information and Communication Engineering (ICEICE)* (Foundation of Computer Science, New York, 2011), pp. 1446–1449.

3.22. Wang C., Mueller F., Engelmann C., Scott S.L. Proactive process-level live migration in HPC environments// *Proc. of the IEEE/ACM Int. Conf. on High Performance Computing, Networking, Storage and Analysis*, Austin, TX, USA, 2008, pp. 1-12.

3.23. Hammerschmidt B.C., Linnemann V. Migratable Web services: increasing performance and privacy in service oriented architectures// *International Journal on Computer Science and Information Systems*, 2006, 1(1), pp. 42-56.

3.24. Engelmann C., Vallee G.R., Naughton T., Scott S.L. Proactive fault tolerance Using preemptive migration// *Proc. of 17th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing*, 2009 pp. 252-257.

3.25. Chakravorty S., Mendes C.L., Kale L.V. Proactive fault tolerance in MPI applications via task migration// *Lecture Notes in Computer Science*, 2006, Vol. 4297, pp. 485-496.

3.26. Fu S., Xu C. Service migration in distributed virtual machines for adaptive grid computing// *Int. Conf. on Parallel Processing*, 2005, pp. 358-365.

3.27. Wood T., Ramakrishnan K.K., van der Merwe J. CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines// *Proc. of 7th ACM SIGPLAN/SIGOPS Int. Conf. on Virtual Execution Environments (VEE)*, 2011, pp. 121-132.

3.28. Choi B.K., Rho S., Bettati R. Fast software component migration for

applications survivability in distributed real-time systems// Proc. of 7th IEEE Int. Symp. on Object-Oriented Real-Time Distributed Computing, Vienna, Austria, 2004, pp. 269-276.

3.29. Zuo Y., Wang X. Semantic specification of service migration for system survivability// Proc. of IEEE Int. Conf. on Electro/Information Technology, Milwaukee, USA, 2014.

3.30. Aglet user manual. <http://aglets.sourceforge.net/>.

3.31. Aglets. <http://en.wikipedia.org/wiki/Aglets>.

3.32. Ishikawa F., Yoshioka N., Tahara Y., Honiden S. Mobile agent system for Web services integration in pervasive networks// Proc. of the 1st Int. Workshop on Ubiquitous Computing, 2004, pp. 38–47.

3.33. Brazier F.M.T., Overeinder B.J., Steen M.V., Wijngaards H.J.E. Generative mobile agent migration in heterogeneous environments// Proc. of the AIMS Workshop at SAC 2002, 2002, pp. 101-106.

3.34. Higashino M., Takahashi K., Kawamura T., Sugahara K. Mobile agent migration based on code caching// Proc. of 26th Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA), 2012, pp. 651-656.

3.35. Neogy R., Chowdhury C., Neogy S. Reliability of mobile agents for reliable service discovery protocol in MANET// International Journal of Wireless & Mobile Networks, 2011, Vol.3, No. 5, pp. 229-243.

3.36. Zgaya H., Hammadi S. Multi-agent information system using mobile agent negotiation based on a flexible transport ontology// Proc. of the Sixth Int. Joint Conf. on Autonomous Agents and Multi-Agent System, Hawaii, USA, 2007, pp. 1024-1036.

3.37. Zhang J. A mobile agent-based tool supporting Web services testing// Wireless Personal Communications, 2011, 56, pp. 147-172.

4. Проектирование архитектур сложных многоагентных систем

4.1. Увеличение срока службы сети в многоагентных системах на основе построения деревьев агрегации данных

4.1.1. Постановка задачи и начальные условия

В этом разделе излагается постановка задачи и описываются начальные условия.

Структура DAT

В работе сеть моделируется как граф $G = (V, E)$. Узлы-датчики представлены в виде вершин V графа. Узлы, являющиеся источниками генерации данных, обозначаются как множество S , узлы, являющиеся ретрансляторами данных, обозначаются как множество R . E является множеством линий связи в сети. Линия e , где $e \in E$, между двумя узлами означает, что эти два узла находятся в пределах дальности связи друг с другом. Работа, описываемая в этой главе, предполагает отсутствие связей с потерями и отказов узлов в сети, чтобы продемонстрировать жизнеспособность предложенной работы.

DAT T для сети G представляется как $T = (V, D)$ так, что $D \in E$. Узлы V и ребра D в T определяют структуру DAT. Каждый узел в T имеет множество входящих ребер I_c от своих дочерних узлов и исходящее ребро O_p к своему родительскому узлу. Структура DAT определяется объединением множества I_c и O_p для всех узлов. Пусть N – количество узлов в WSN, тогда структура DAT является множеством ребер D , заданным как

$$D = O_p^1 \cup I_c^1 \cup O_p^2 \cup I_c^2 \cup \dots \cup O_p^N \cup I_c^N \quad (4.1)$$

Коэффициент DA α

Требования приложения определяют, как и когда узлы передают данные. В приложениях для мониторинга и наблюдения данные собираются периодически с интервалом времени t со всех узлов в сети. Сценарий благоприятствует использованию метода DA для снижения накладных расходов при передаче данных. Уменьшение количества передач контролируется коэффициентом DA.

Определение. Коэффициент DA α – это количество блоков данных, объединенных в один пакет [4.4, 4.6].

Каждый узел источника u генерирует данные d^g , принимает данные d^i , объединяет их с использованием α и передает пакеты $u.P^T$, определяемые как

$$u.P^T = \left\lceil \frac{d^i + d^g}{\alpha} \right\rceil \quad (4.2)$$

Каждый ретрансляционный узел u принимает данные d^i и передает пакеты $u.P^T$, определяемые как

$$u.P^T = \left\lceil \frac{d^g}{\alpha} \right\rceil \quad (4.3)$$

Время жизни сети

NL – это время, в течение которого все узлы в сети могут передавать данные. Для улучшения NL увеличивается время смерти первого узла. Узел умирает, когда его энергетический запас истощается и он не может передавать данные в приемник. В DAT узлы получают данные от дочерних узлов, агрегируют их и отправляют на родительский узел. Процесс продолжается до тех пор, пока данные со всех узлов не достигнут приемника за единицу времени t . Продолжительность интервала для t определяется приложением и является одним циклом DA. Общее количество циклов DA, для которых все узлы в сети могут передавать

данные, является NL. Цикл DA обозначается как DAR.

Вычисление NL

NL основано на энергии, потребляемой узлами в сети. В работе модель энергопотребления учитывает только энергию, потребляемую при передаче данных, поскольку энергия, потребляемая при вычислениях и измерениях, пренебрежимо мала по сравнению с передачей данных [4.1]. Пусть энергия, потребляемая узлом $SeedN$, где $SeedN \in V$, в одном DAR обозначается как $SeedN.En$. Пусть ε , Tx и Rx – доступная энергия узла, энергия, необходимая для передачи пакета, и энергия, необходимая для приема пакета, соответственно. Энергия E_R , потребляемая при получении данных, является суммой энергий, необходимой для приема пакетов данных от дочерних узлов. Пусть $CD = \{cid_1, cid_2, \dots, cid_{cn}\}$ является множеством дочерних узлов $SeedN$, тогда E_R определяется как

$$SeedN.E_R = (cid_1.P^T + cid_2.P^T + \dots + cid_{cn}.P^T) \times Rx \quad (4.4)$$

Энергия E_T , потребляемая $SeedN$ при передаче данных, определяется как

$$SeedN.E_T = SeedN.P^T \times Tx \quad (4.5)$$

Используя уравнения (4.4) и (4.5), полная энергия En , потребляемая узлом $SeedN$ за один цикл, равна

$$SeedN.En = SeedN.E_T + SeedN.E_R \quad (4.6)$$

Время жизни узла $SeedN$ – это количество DAR, в течение которых узел может передавать данные, рассчитанное как

$$SeedN.nl = \frac{SeedN.\varepsilon}{SeedN.En} \quad (4.7)$$

Используя уравнения (4.2)-(4.7), каждый узел вычисляет свое значение $SeedN.nl$. Время жизни сети NL – это время жизни узла с наименьшим значением $SeedN.nl$, заданное как

$$NL = \min\{SeedN_1.nl, SeedN_2.nl...SeedN_N.nl\} \quad (4.8)$$

Сценарии для NL

Более высокие (низкие) значения α приводят к меньшему (большему) количеству передаваемых пакетов, как это определяется уравнением (4.2), и эффективно увеличивают (уменьшают) NL. Определяя границы для α , NL точно настраивается в соответствии с требованиями приложения. Пусть $\min \alpha$ и $\max \alpha$ представляют минимальные и максимальные значения для α . Для $\alpha = \min \alpha = 1$ все блоки данных $(d^i + d^s)$ сообщаются узлами индивидуально. Этот сценарий представляет наихудший случай для NL. Для $\alpha = \max \alpha$ максимальное количество блоков данных объединяется в один пакет. Чем выше значение $\max \alpha$, тем больше блоков данных можно объединить, что приведет к улучшению NL. Этот сценарий представляет собой лучший случай для NL. Для всех других значений α , таких что $\min \alpha < \alpha < \max \alpha$, NL ограничивается наихудшими и наилучшими значениями.

Графы DAT

Требования к приложениям определяют, какой DAT является оптимальным для данного сценария. Однако найти оптимальный DAT – сложная проблема. Поэтому разрабатываются и реализуются алгоритмы для поиска неоптимальных DAT, которые отвечают требованиям приложений. В предлагаемой работе последовательность DAT, которая удовлетворяет требованиям приложения, идентифицируется и представляется в виде графа $U = [DATs, DARs]$, где $DATs = \{T_1, T_2, \dots, T_j\}$ и $DARs = \{NL_1, NL_2, \dots, NL_j\}$. $\{T_1, T_2, \dots, T_j\}$ является последовательностью DAT, а $\{NL_1, NL_2, \dots, NL_j\}$ является их соответствующими NL. Граф показывает, что DAT T_1 соответствует DAR NL_1 , DAT T_2 соответствует для DAR NL_2 и т.д. Длина графа, обозначенная как U_{len} , является количеством DAT в графе.

Большая длина означает большее количество DAT, что, в свою очередь, означает более высокую стоимость реструктуризации DAT в сети.

4.1.2. Постановка проблемы DATRL

Дано:

1. Граф $G = (V, E)$, представляющий сеть.
2. $V = \{S \cup R \cup s\}$, где S – множество источников, R – множество узлов ретрансляции, а s – приемник. $|V| = |S| + |R| + 1 = N + 1$.
3. E – множество каналов связи в сети.
4. $d^s \in Z^+$ для каждого источника $v \in S$ и $d^s = 0$ для $v \in R$.
5. $\varepsilon, Tx, Rx \in R^+$ представляют доступную энергию узла, энергию, потребляемую для передачи одного пакета, и энергию, потребляемую для приема одного пакета, соответственно.
6. α представляет коэффициент DA, такой, что $\min \alpha \leq \alpha \leq \max \alpha$, где $\min \alpha, \alpha, \max \alpha \in Z^+$.

Цель задачи DATRL состоит в том, чтобы найти граф $U = [\{T_1, T_2, \dots, T_j\}, \{NL_1, NL_2, \dots, NL_j\}]$, который максимизирует NL и минимизирует длину графа, определяемую как $Max \sum_{i=1}^j NL_i, Min j$

Минимизация j означает меньшее значение $Ulen$ и снижает стоимость реструктуризации соответствующего дерева. Каждое DAT T_i , где $i = [1..j]$, использует подход ELV.

4.1.3. Подход ELV

Предлагаемый подход ELV (рис. 1.1в) облегчает применение в агрессивных средах, в которых, как правило, сенсорные узлы не являются ни заменяемыми, ни перезаряжаемыми. В таких ситуациях ELV балансирует между глобальными и локальными методами построения

DAT. В этом разделе описывается подход ELV, который использует информацию отдельных узлов, называемую локальной информацией (LI) каждого узла, и информацию приемника, называемую расширенной локальной информацией (ELI) узла-приемника.

Локальная информация LI

В DAT каждый узел имеет путь связи с приемником. Самостоятельно узел не определяет полный путь к приемнику, но имеет контроль над путями, сформированными в его окрестности. Пусть LI в узле *SeedN* будет представлена его атрибутами. Подробная информация об атрибутах приведена в табл. 4.1 вместе с состоянием узла при заполнении соответствующего атрибута. Подробное описание состояний узлов представлено в разделе 4.1.4.

Таблица 4.1

Атрибуты узла *SeedN* и его состояния при заполнении соответствующего атрибута

Атрибут узла	Описание	Состояние узла
<i>id</i>	идентификатор	$q0$
ε	доступная энергия	$q0$
<i>ND</i>	информация о соседних узлах	$q1$
<i>pn</i>	родительский узел	$q1$
<i>hc</i>	количество переходов до приемника	$q1$
α	коэффициент DA	$q1$
d^g	генерируемые данные	$q1$
<i>CD</i>	информация о дочерних узлах	$q3$
d^i	входящие данные	$q3$
<i>nl</i>	время жизни	$q3$
<i>En</i>	энергия, потребляемая за один цикл	$q3$

P^T	количество переданных пакетов	q^3
P^R	количество полученных пакетов	q^3

ND и CD являются составными атрибутами. ND содержит идентификаторы узла nid , счетчик переходов hc и время жизни nl всех соседних узлов. Пусть nn – количество соседних узлов $SeedN$, тогда ND описывается как

$$ND[1] = \{nid_1, hc_1, nl_1\}, \dots, ND[nn] = \{nid_{nn}, hc_{nn}, nl_{nn}\} \quad (4.9)$$

CD содержит идентификаторы cid всех дочерних узлов. Пусть cn будет числом дочерних узлов $SeedN$, тогда

$$CD = \{cid_1, cid_2, \dots, cid_{cn}\} \quad (4.10)$$

Расширенная локальная информация ELI

ELI – это доступная в приемнике информация об узлах, которые находятся на расстоянии k переходов от приемника. Значение k зависит от приложения. Используя ELI, приемник применяет методы оптимизации для реструктуризации DAT.

4.1.4. Алгоритм ELVF

В этом разделе описывается предложенный алгоритм ELVF. На рис. 4.2 показан сценарий применения ELVF.

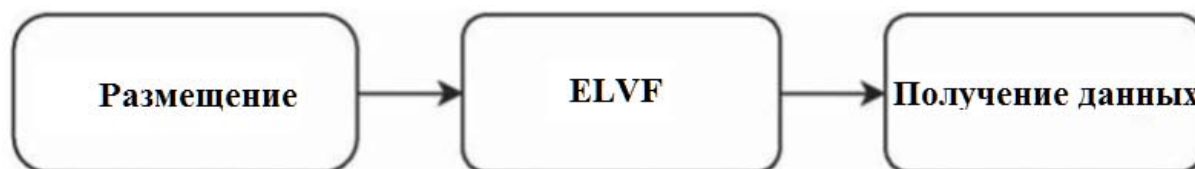


Рис 4.2. Сценарий применения ELVF

На рис. 4.3 представлены этапы ELVF:

1. Размещение: размещение N узлов.
2. Начальное построение DAT: начальное построение DAT T_{INIT} с использованием LI. Этот шаг выполняется каждым узлом в сети.
3. Реструктуризация DAT и составление графа U : Этот шаг выполняется в приемнике с использованием его информации о k переходах.
 - 3.1. Часть T_{INIT} , которая включает в себя узлы с $hc \leq k$, используется для реструктуризации в T_{LPR} с использованием LPR (на основе LI).
 - 3.2. Реструктуризация T_{LPR} в T_{EPR} с использованием EPR (на основе ELI).
4. Сбор данных: каждый узел отправляет данные своему родительскому узлу, пока они не достигнут приемника. Узлы с $hc \leq k$ отправляют данные в родительские узлы по графу U .



Рис. 4.3. Шаги ELVF

Подробное описание шагов ELVF представлено далее.

Шаг I: размещение

N датчиков случайным образом размещаются в интересующей области. На этом этапе каждому узлу назначается идентификатор id . Все

узлы находятся в состоянии $q0$, как показано на рис. 4.4.

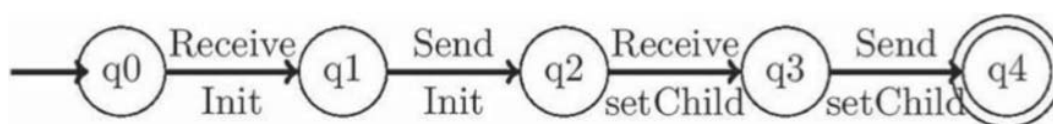


Рис. 4.4. Схема установления начального пути

Шаг II: начальное построение DAT

На уровне сети приемник инициирует построение DAT, путем широковещательной передачи сообщения $initMsg(sendernode = sink, hopcount = 0)$. Узлы в диапазоне передачи приемника получают это сообщение, обновляют его и отправляют соседним узлам. Процесс продолжается до тех пор, пока все узлы не получают сообщение и не будет создано DAT.

На уровне узла каждый узел заполняет свои атрибуты LI, представленные в табл. 4.1. На этом этапе $SeedN$ обнаруживает информацию ND о своих соседних узлах, определяет количество переходов hc , а также выбирает и устанавливает путь к своему родительскому узлу pn и дочерним узлам CD . Установление пути на уровне узла облегчается тем, что каждый узел циклически проходит через состояния приема, вычисления и передачи. Начальное установление пути на уровне узла представлено на рис. 4.4.

Первоначально каждый узел находится в состоянии $q0$, которое является состоянием прослушивания / приема. Находясь в $q0$ $SeedN$ получает $initMsg$ от соседних узлов и переходит в состояние $q1$.

В состоянии $q1$ $SeedN$ обновляет свой ND , как показано уравнением (4.9) с помощью $\{nid = sendernode, hc = hopcount\}$ для nn соседних узлов. $SeedN$ проверяет ND для идентификации соседнего узла x с наименьшим значением hc , и назначает его в качестве родительского узла, установив $SeedN.pn = x$ и $SeedN.hc = x.hc + 1$. Назначение узла с наименьшим значением

hc в качестве родительского узла устанавливает кратчайший путь к приемнику. $SeedN$ обновляет $initMsg$ до $initMsg(sendernode = SeedN.id, hopcount = SeedN.hc)$, отправляет его своим nn соседним узлам и переходит в состояние $q2$. В состоянии $q2$ $SeedN$ снова находится в режиме приема и получает $setChildMsg(id)$ от своих дочерних узлов и переходит в состояние $q3$. В состоянии $q3$ $SeedN$ обновляет CD с $cid = id$ для cn дочерних узлов и вычисляет $SeedN.En$, используя уравнения (10) и (6) соответственно. $SeedN$ отправляет сообщение $setChildMsg(SeedN.id)$ своему родительскому узлу $SeedN.pn$ и переходит в состояние $q4$. Состояние $q4$ означает, что узел собрал LI и установил начальный путь к приемнику.

Когда все узлы в сети устанавливаются в состояние $q4$, создается начальное DAT T_{INIT} .

Шаг III: реструктуризация DAT

В ELVF узлы в пределах расстояния k переходов от приемника имеют $hc \leq k$. Узлы за пределами k переходов имеют $hc > k$. Приемник использует LPR и EPR для определения графа DAT как показано на рис. 4.5.

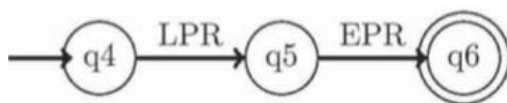


Рис. 4.5. Схема реструктуризации пути

Шаг IV: сбор данных

Конечные узлы получают данные и отправляют их на родительский узел. Все промежуточные узлы получают данные от дочерних узлов и отправляют их родительским узлам. Количество пакетов, отправленное узлом, рассчитывается с помощью формул (4.2) и (4.3).

4.1.5. Методы LPR и EPR

В этом разделе рассматриваются методы LPR и EPR предложенного алгоритма ELVF. LPR основан на эвристике, применяемой каждым узлом. EPR использует жадный алгоритм и информацию о количестве k переходов для реструктуризации путей в DAT. Рассмотрим сеть, показанную на рис. 4.6.

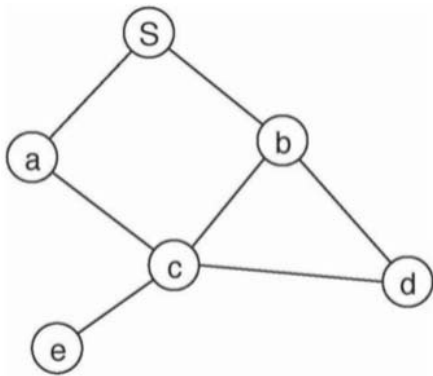


Рис. 4.6. WSN с вершинами $a-e$ и источником S.

Метод LPR

LPR – это метод, основанный на LI для данного $SeedN$. Он основан на эвристике, что, если время жизни узла увеличивается с помощью LI, тогда даже в глобальном масштабе сеть может выжить для большего количества DAR. После первоначального построения дерева каждый узел в сети находится в состоянии $q4$. Применяя LPR, узел переходит в состояние $q5$, как показано на рис. 4.5. Шаги LPR, выполняемые в узле $SeedN$, следующие:

1. Каждый узел анализирует свои данные ND и используя значения $nid.nl$ определяет, должен ли он переключиться на новый родительский узел new_pn . Порядок, в котором осуществляется доступ к значениям $nid.nl$ узлов в ND является критическим параметром для увеличения NL. В ELVF доступ к узлам в ND осуществляется в порядке убывания их

значений $nid.nl$. Убывающий порядок подразумевает, что узлы с большей доступной энергией ε рассматриваются в первую очередь как возможные варианты переключения родителя.

2. Если переключение улучшает $new_pn.nl$ по сравнению с текущим $SeedN.pn.nl$, то переключение выполняется. В противном случае $SeedN.pn$ остается без изменений.

Метод EPR

Метод EPR основан на ELI и увеличивает NL, используя жадный алгоритм. В EPR увеличивается время жизни узла с наименьшим временем жизни. Алгоритм состоит из следующих шагов:

1. Определяется узел x , который имеет минимальное значение $SeedN.nl$.

2. Проверяется $x.CD$, чтобы узнать, можно ли переключить какой-либо узел в $x.CD$ на новый родительский узел, чтобы улучшить $x.nl$. Такое переключение называется благоприятным переключением. Если обнаружено благоприятное переключение, то оно производится и осуществляется возврат на предыдущий шаг. Иначе в $x.CD$ ищется узел с минимальным значением nl и определяется как xt . Назначается $x = xt$ и повторяется текущий шаг, пока не будут достигнуты узлы на расстоянии k .

3. Создание благоприятного переключения улучшает $x.nl$ и эффективно увеличивает время жизни сети NL.

Пример

Рассмотрим сеть с пятью узлами и приемником, как показано на рис. 4.6. Пусть начальная энергия ε для всех узлов составляет 10 единиц. Пусть T_x и R_x имеют значения 2 единицы и 1 единицу соответственно. Предложенный алгоритм ELVF создает граф DAT, как показано на рис.

4.7. В графе T_1 собираются данные для двух DAR. После EPR идентифицирует узел b как узел с минимальным временем жизни и реструктурирует DAT до T_2 . В T_2 узел b становится конечным узлом. T_2 живет еще один цикл. Таким образом, $NL=3$ и $j=2$, поскольку созданы два DAT.

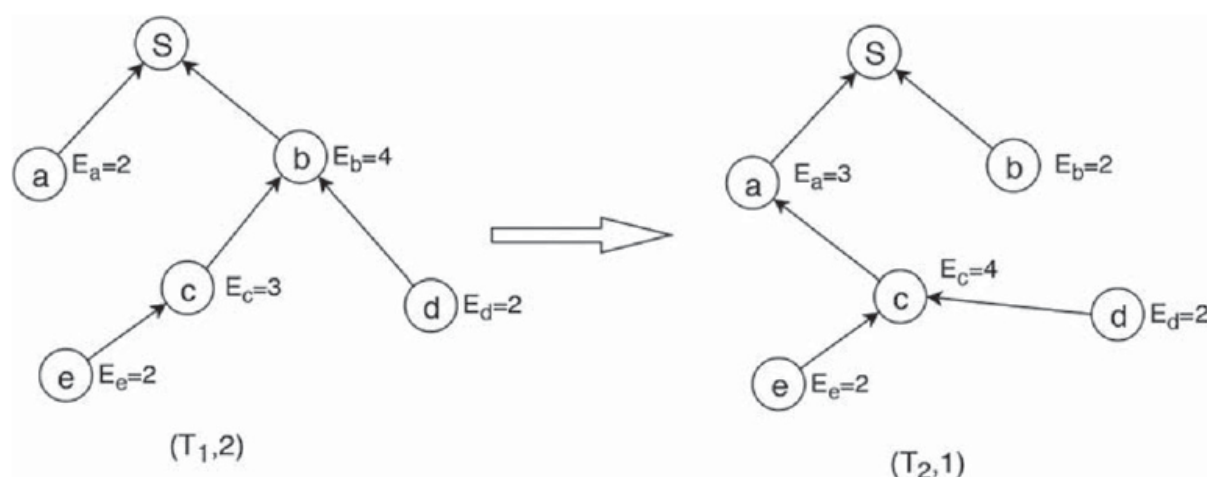


Рис. 4.7. Граф $U = \{(T_1, T_2)\}, \{(1, 2)\}$; $NL=3, j=2$

Алгоритм ELVR

ELVR схож с ELVF за исключением того, что узел приемника не является фиксированным. В ELVR любая случайная позиция в интересующей области выбирается как позиция приемника. Размещение приемника случайным образом желательно в сценариях наблюдения. Приемник в фиксированном положении может быть легко атакован. Однако случайно расположенный приемник повышает сложность сценария, затрудняя его идентификацию злоумышленником, что резко снижает вероятность того, что на приемник будет совершена атака.

4.1.6. Оценка эффективности

Параметры моделирования

При настройке моделирования WSN генерируются путем расстановки узлов ретрансляции и узлов-датчиков в интересующей

области. Для облегчения сравнения производительности, настройка моделирования аналогична настройке LTRBSA [4.6].

N узлов устанавливаются случайным образом, где $|N|$ варьируется между значениями 100 и 500 с шагом 100. Расстановка осуществляется в поле 10×10 единиц с дальностью передачи $g=2$ для каждого узла. Приемник расположен в позиции (5,10) в интересующей области. Количество узлов ретрансляции $|R|=50$. Начальная энергия каждого узла $SeedN.\varepsilon = 10^5$ единиц, энергия Tx , необходимая для передачи пакета, равна 2 единицам, а энергия Rx , необходимая для приема пакета, равна 1 единице. Количество данных d^g , генерируемое узлами, выбирается случайным образом в интервале от 1 до B , где $B=10$ единиц. Коэффициент DA α изменяется между $\min \alpha$ и $\max \alpha$, где $\min \alpha = 1$ и $\max \alpha = 7$. Значение $k=3$. Для каждого значения $|N|$ генерируется 30 сетей, и наблюдаются значения NL.

Наблюдение за NL

Рассмотрим сеть из 100 случайно расположенных узлов в интересующей области, показанную на рис. 4.8. Узел-приемник выделен окружностью.

На рис. 4.9 показаны значения энергии, потребляемой узлами, которые находятся на расстоянии k переходов от приемника. На графике узлы обозначены их идентификатором, а энергия, потребляемая на каждом цикле DA, равна $id.En$. Значения потребляемой энергии при моделировании были приведены к диапазону от 0 до 1.

Как видно, максимальная энергия потребляется узлом с $id = 32$ при использовании LTRBSA. Также видно, что для алгоритма ELVF максимальная энергия потребляется тем же узлом. Однако, при ELVF узел потребляет только 62% энергии, от потребляемой с использованием

LTRBSA. Результаты показывают, что предложенный алгоритм ELVF уравнивает энергопотребление узлов.

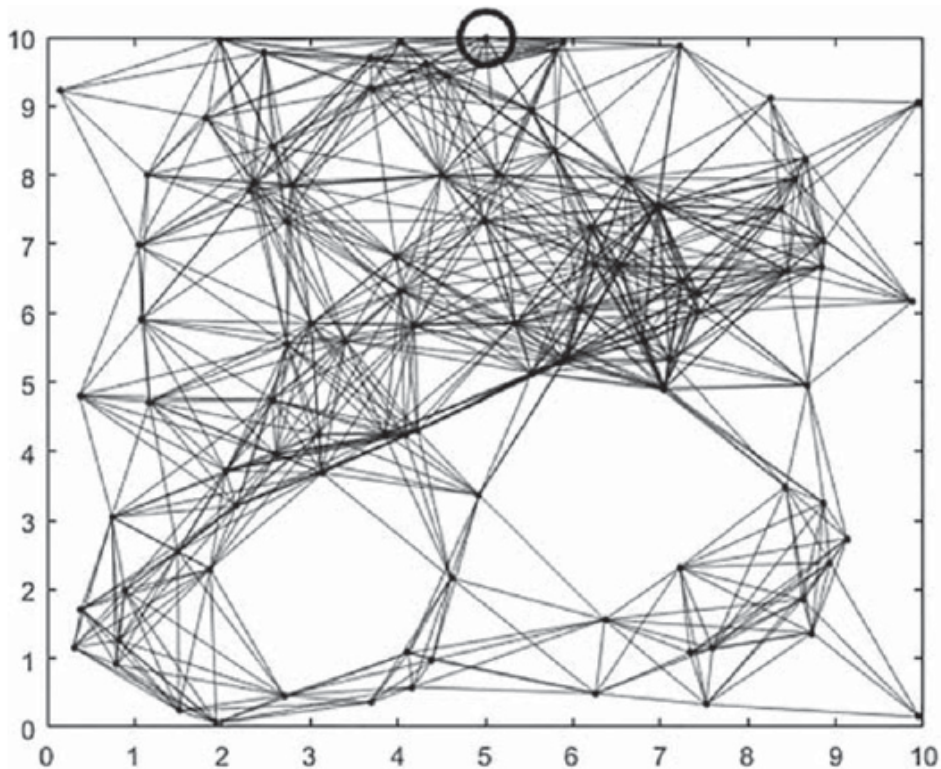


Рис. 4.8. Пример сети со 100 случайно расположенными узлами

Значения NL для алгоритмов ELVF, ELVR и LTRBSA сравниваются на рис. 4.10 для количества датчиков от 100 до 500. Результаты показывают, что ELVF увеличивает NL на 6-40% по сравнению с LTRBSA для количества узлов N в диапазоне от 100 до 500. Эти наблюдения показывают, что предложенный алгоритм ELVF обладает высокой масштабируемостью.

Для большего понимания мы рассчитали стандартное отклонение значений энергопотребления узлов в сети, что показано на рис. 4.11. Каждая точка данных при моделировании была усреднена за 30 прогонов и приведена к диапазону от 0 до 1 при $\alpha = 3$. Как видно, предложенные алгоритмы уравнивают энергопотребление и приводят к увеличению NL.

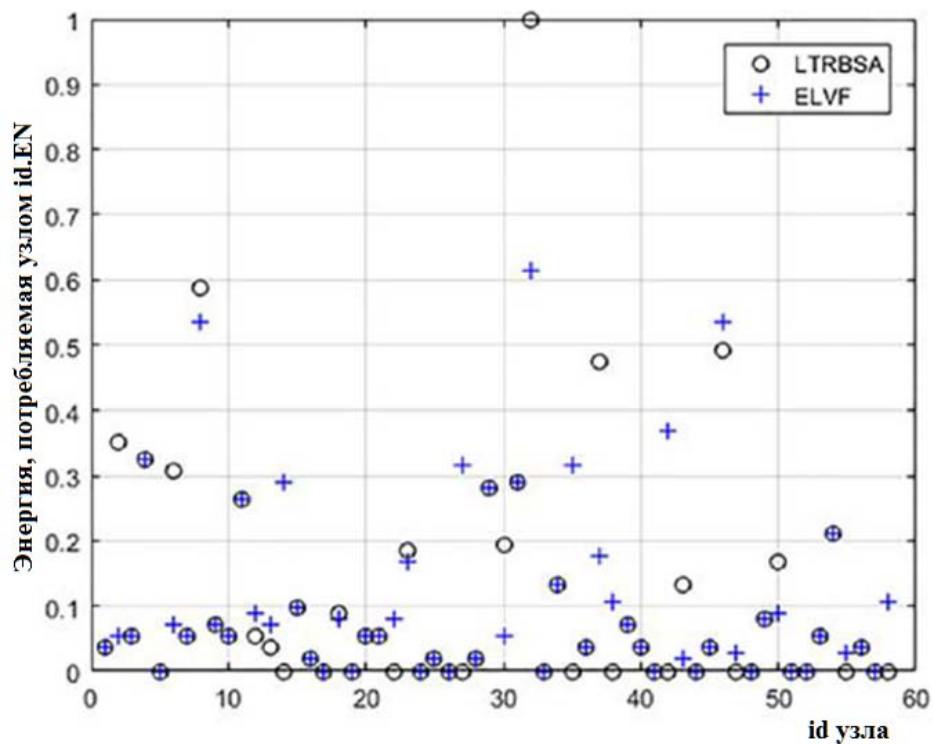


Рис. 4.9. График значений энергии, потребляемой узлами

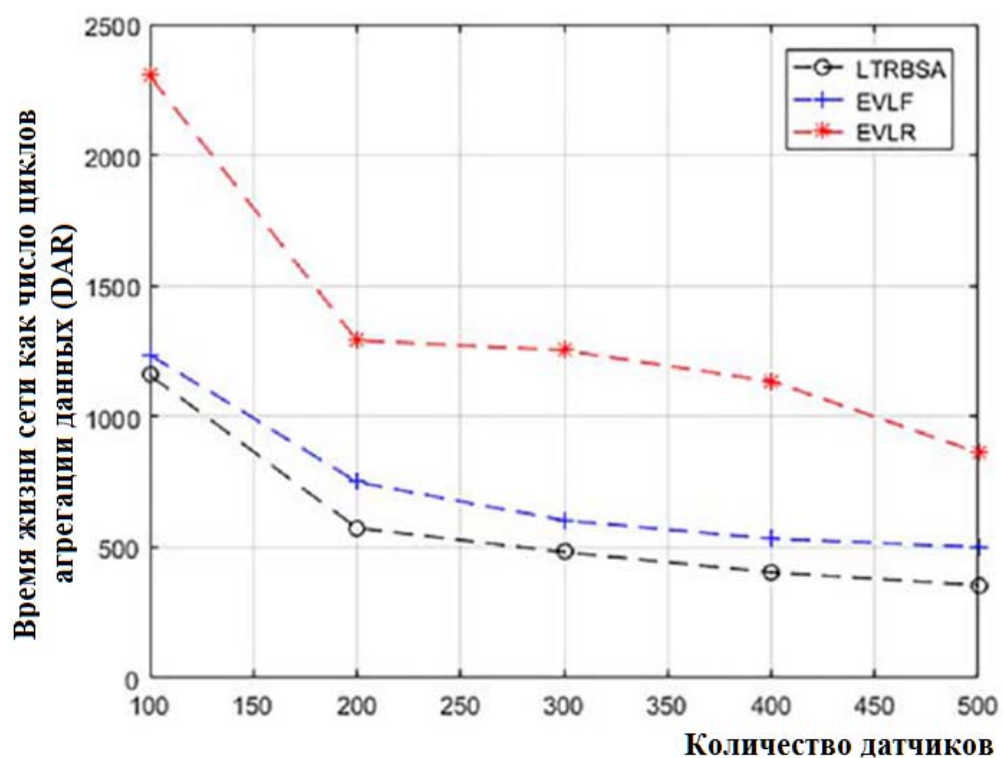


Рис. 4.10. Время жизни сети в терминах числа циклов агрегации данных (DAR) для числа датчиков от 100 to 500 ($\alpha=3$)

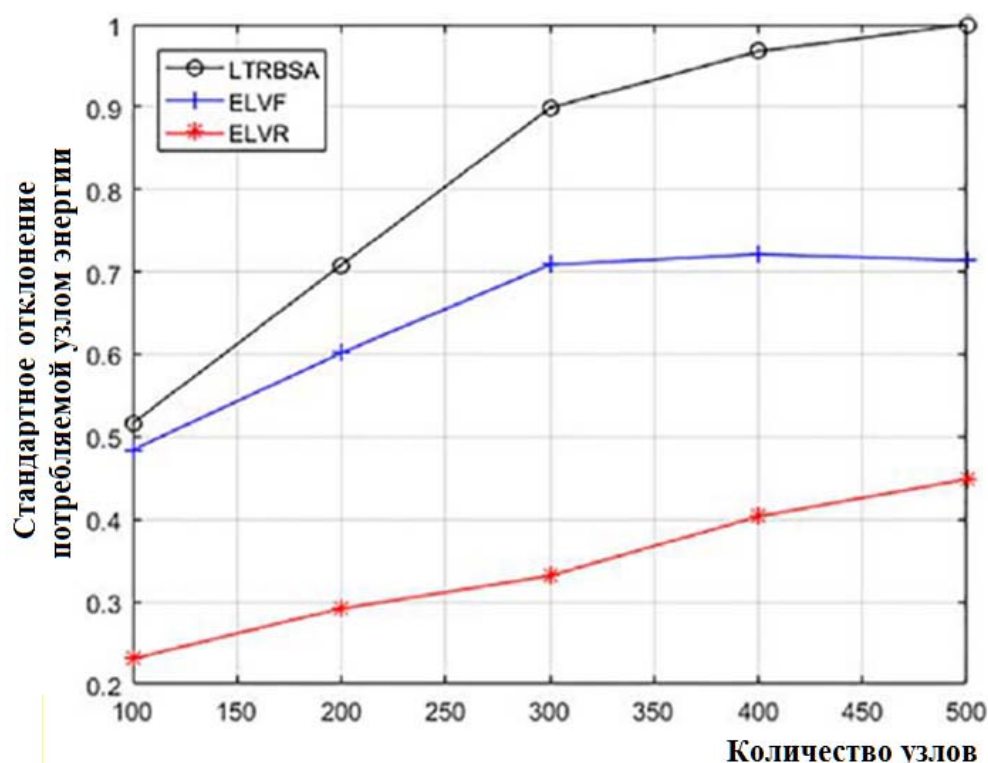


Рис. 4.11. Стандартное отклонение значений энергопотребления

Наблюдение за коэффициентом $DA \alpha$

Значения NL для алгоритмов ELVF, ELVR и LTRBSA сравниваются для коэффициентов агрегации $\alpha = 1, 3$ и 7 на рис. 4.12-4.14 соответственно. Наблюдается, что NL увеличивается с увеличением значения коэффициента α .

Процент увеличения NL при использовании ELVF по сравнению с LTRBSA составляет 6% для 100 узлов независимо от значения α . Тем не менее, по мере увеличения количества узлов, процент растет. Из результатов моделирования, полученных для 500 узлов, видно, что при использовании ELVF увеличение NL составляет 47% при $\alpha = 1$ и 40% и 33% при $\alpha = 3$ и $\alpha = 7$ соответственно. Результаты демонстрируют масштабируемость ELVF для различных значений α .

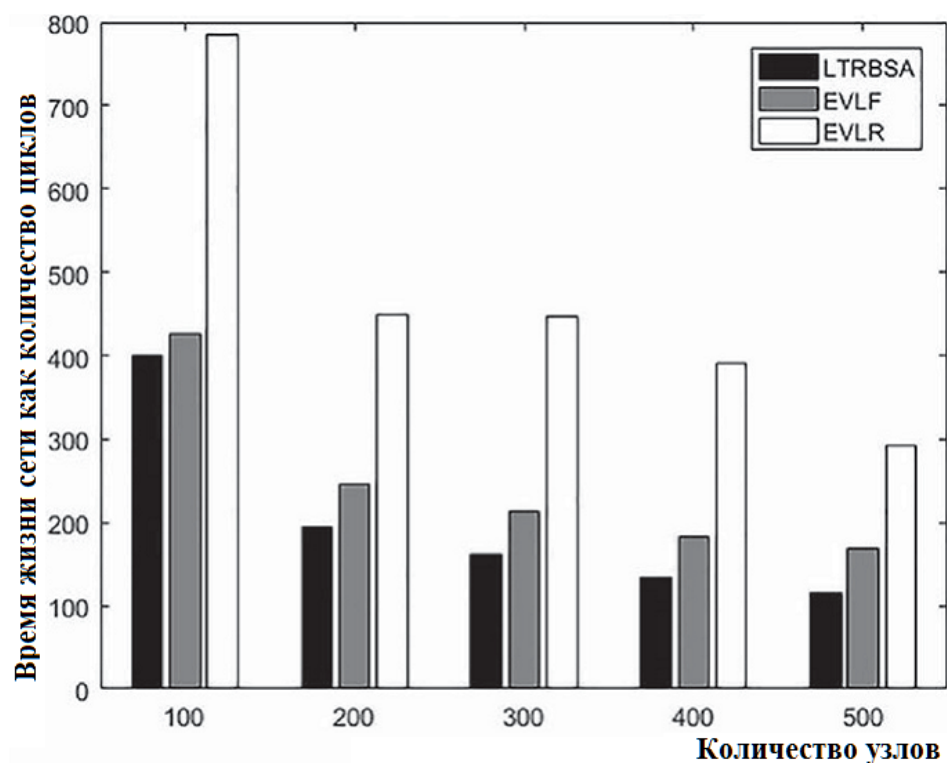


Рис. 4.12. NL при $\alpha = 1$

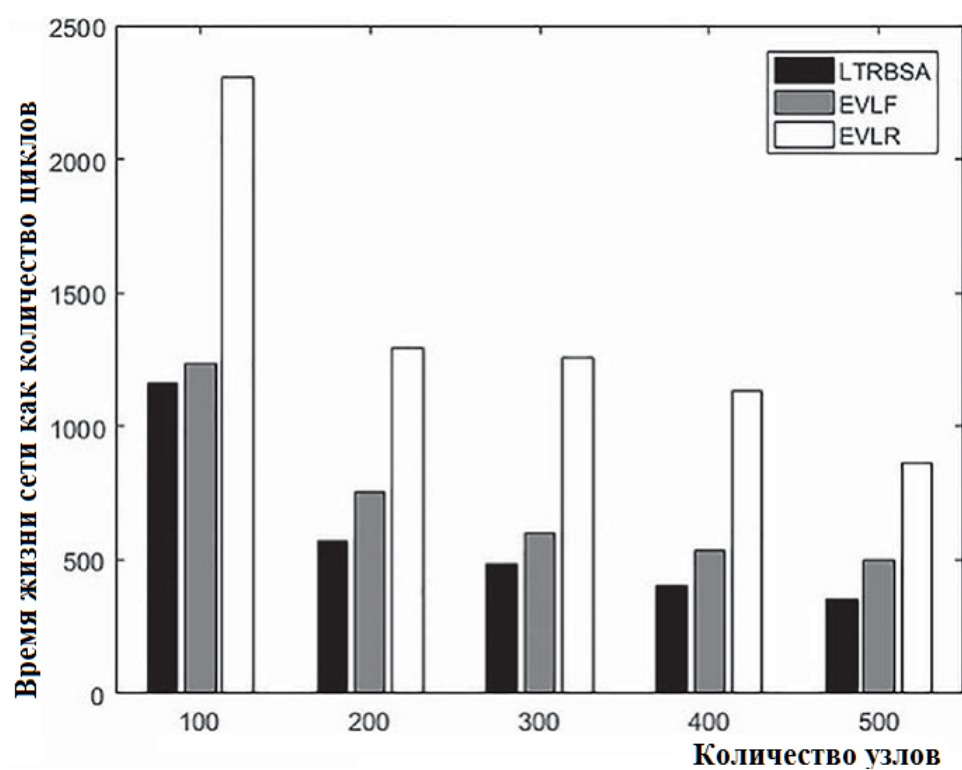


Рис. 4.13. NL при $\alpha = 3$

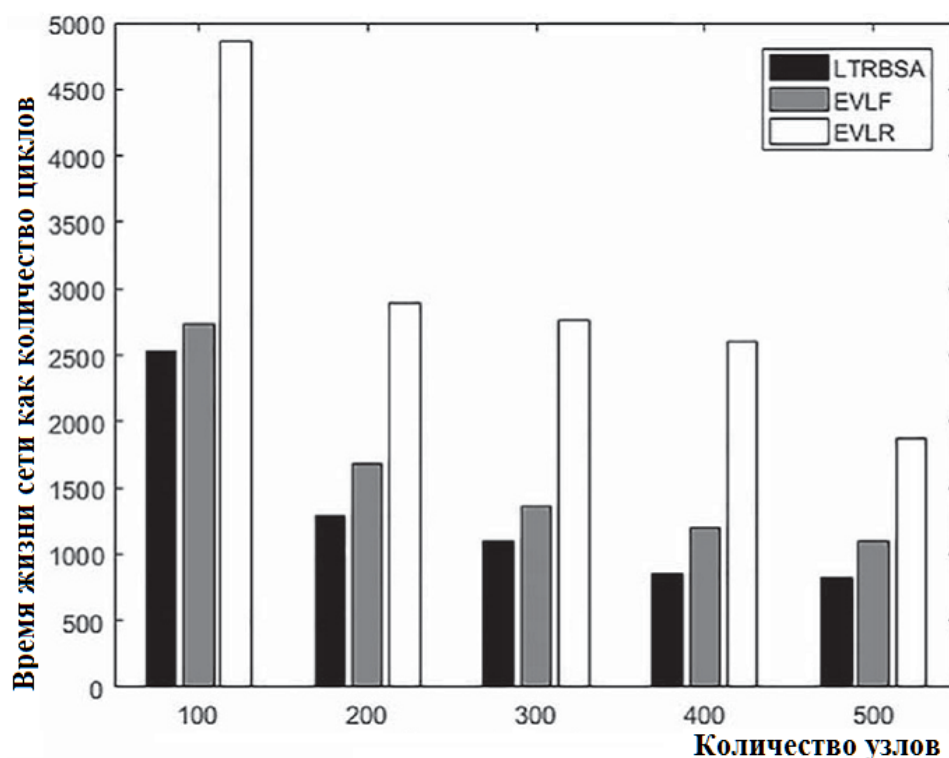


Рис. 4.14. NL при $\alpha = 7$

Наблюдение за количеством DAT

Количество DAT $U.len$ для алгоритмов ELVF и ELVR и LTRBSA сравнивается с использованием гистограмм на рис. 4.15. Значения частоты получены за 100 прогонов для каждого значения α при $N = 30$. Результаты показывают, что $U.len$ колеблется от одного до трех.

Эти результаты показывают, что ELVF и ELVR создают один DAT в 66% прогонов по сравнению с 60% при использовании LTRBSA при $\alpha = 3$ (рис. 15а-в). Как показано на рис. 15г-е, при $\alpha = 7$ ELVF и ELVR создают один DAT в 78% прогонов по сравнению с 66% при использовании LTRBSA. Эти результаты показывают, что ELVF и ELVR идентифицируют более короткие графы, эффективно сокращая накладные расходы на реструктуризацию.

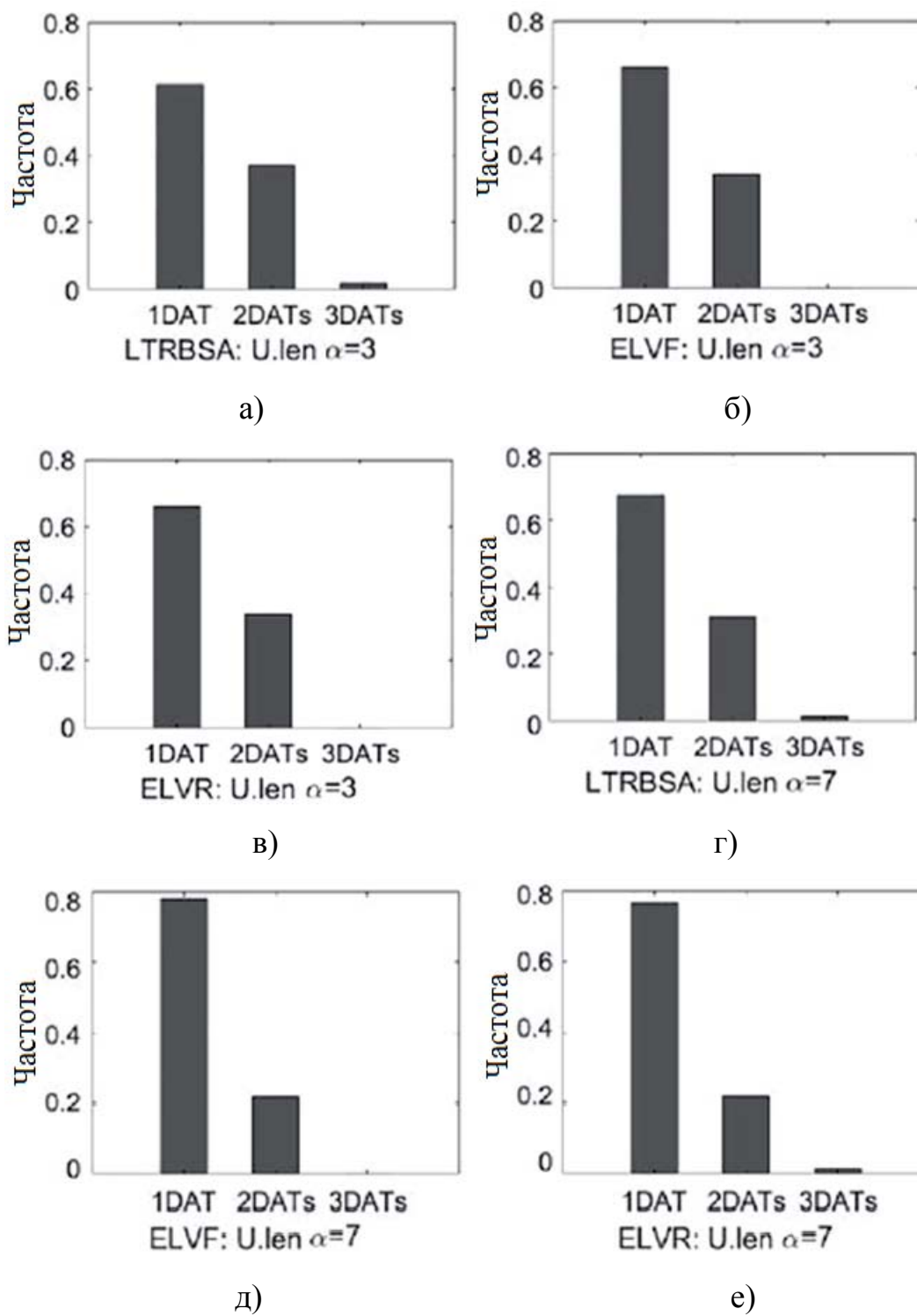


Рис. 4.15. Сравнение $U.len$

Наблюдения за расширяемостью ELVF при использовании ELVR

При использовании ELVR приемник размещается на случайной

позиции. Моделирование показывает, что ELVR увеличивает NL на 70–80% по сравнению с ELVF. Результаты показаны на рис. 4.12–4.14. На рис. 4.11 показано стандартное отклонение значений энергопотребления узлов в сети при использовании ELVR. Эти результаты показывают, что ELVR потребляет меньше энергии по сравнению с ELVF. Увеличение NL в сочетании с уменьшением шансов получить атаку, используя ELVR, демонстрирует расширяемость.

4.2. Высокоуровневое описание архитектуры составных мобильных приложений на основе метамоделирования

4.2.1. Архитектура процесса композиции

В этом разделе представлен процесс гетерогенной композиции и изложены его различные этапы, чтобы помочь в разработке мобильных приложений с учетом контекста (рис. 4.16).

Чтобы представить желаемое составное мобильное приложение (СМА) на архитектурном уровне, на рис. 4.17 предложена соответствующая метамодель, которая обеспечивает высокоуровневое описание архитектуры СМА. С каждой информацией определенный связывается архитектурный элемент, используемый для создания желаемого мобильного приложения. Эта метамодель направлена на определение того, как архитектурные элементы составного приложения связаны друг с другом. Определяя архитектуру СМА, в первую очередь осуществляется фокус на различных функциях желаемого приложения (т.е. требования заказчика), где каждая из них относится к абстрактному объекту программного обеспечения, игнорируя то, как они будут реализованы (например, компоненты, службы, приложения).

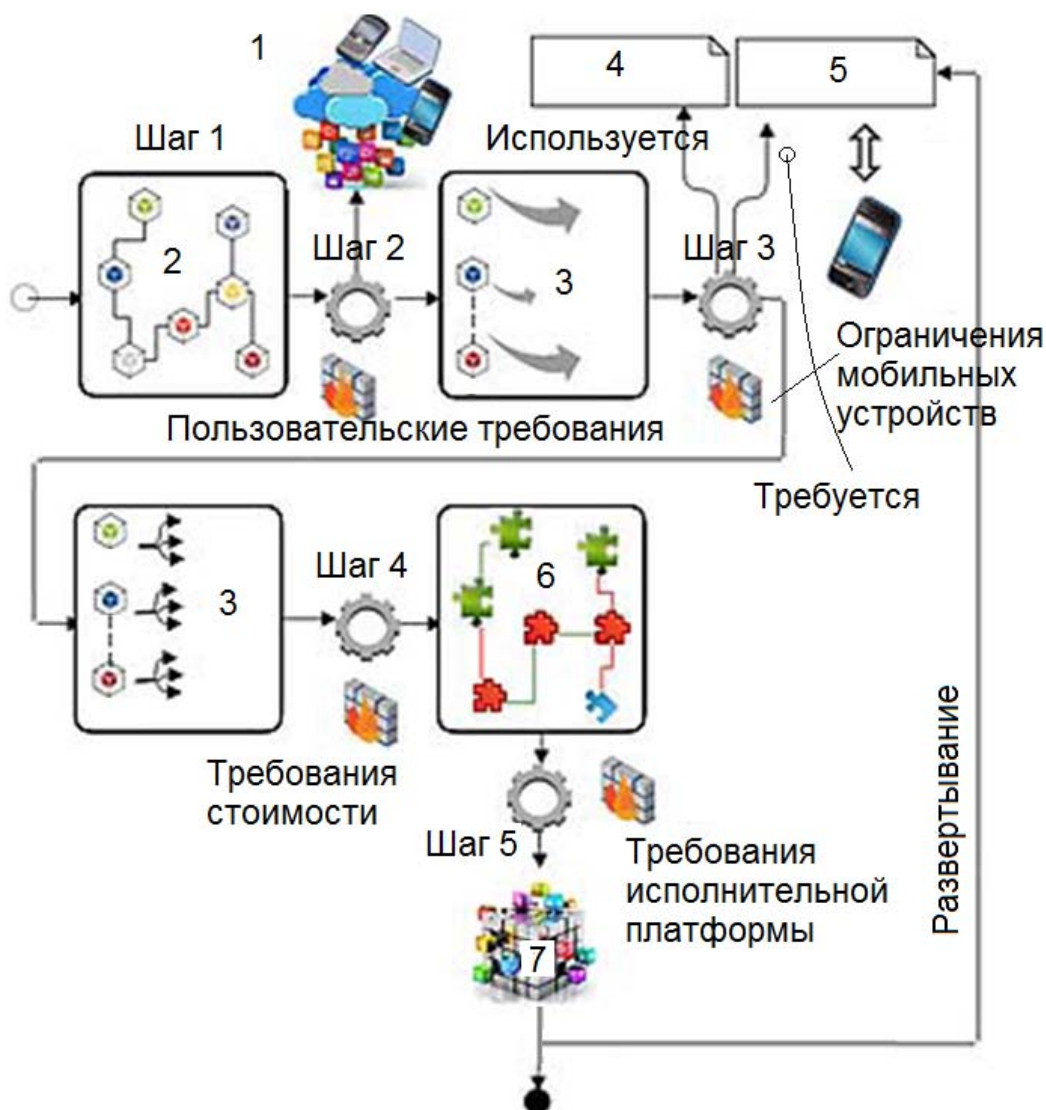


Рис. 4.16. Обзор процесса гетерогенной композиции: 1 – гетерогенные ресурсы; 2 - ожидаемое приложение; 3 – существующие решения; 4 - исполнительные профили существующих решений; 5 – контекст; 6 – абстрактная композиция мобильного приложения; 7 - конкретная композиция мобильного приложения

Во-вторых, необходимо уделить особое внимание подробному описанию нужного приложения. Будем основываться на выборе типа реализации каждого абстрактного программного объекта (т.е. каждой функциональности, описанной в первой архитектуре) для построения этой подробной архитектуры. Этот выбор откладывается до того момента, когда

на шаге 4 предлагаемого процесса будут выбраны конкретные программные объекты. Для этой цели необходимы две следующие архитектуры для описания СМА:

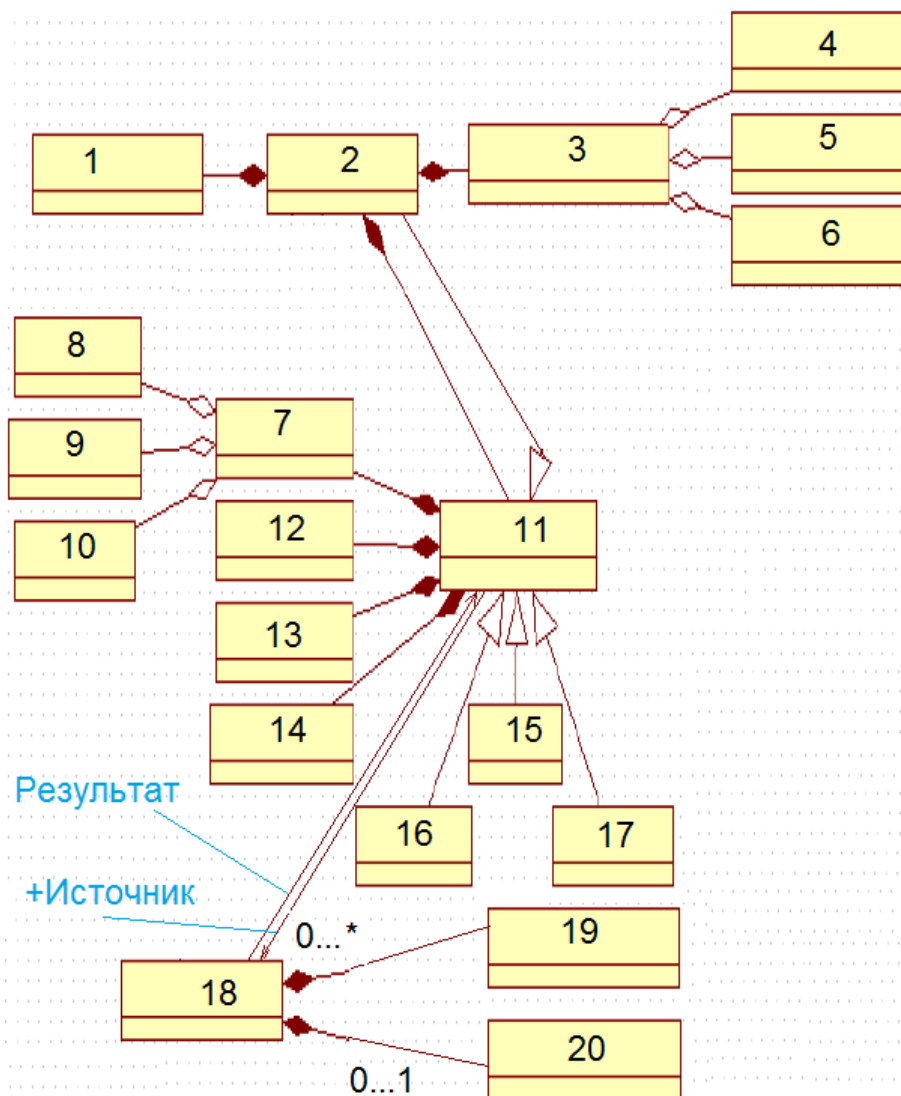


Рис. 4.17. Мета модель СМА: 1 – управление СМА; 2 - СМА; 3 – контекстная информация; 4 – контекст развертывания; 5 – контекст исполнения; 6 - исполнительная платформа; 7 – местоположение источника; 8 – Интернет; 9 – локальный источник; 10 – другие устройства; 11 – программный компонент; 12 – описание; 13 – функциональность; 14 – исполнительный профиль; 15 – приложение; 16 – сервис; 17 – компонент; 18 – отношение; 19 – эндогенный посредник; 20 – экзогенный посредник

1. Абстрактная функциональная архитектура СМА: представляет собой высокоуровневое описание желаемых функций и их зависимостей, используемых для достижения цели композита.

2. Абстрактная подробная архитектура СМА: представляет собой высокоуровневое описание различных конкретных программных объектов, которые будут использоваться для реализации требуемых функций, и различных посредников, которые представляют сотрудничество между повторно используемыми конкретными объектами, и они будут использоваться для устранения неоднородности между этими объектами.

Управление СМА представлено на рис. 4.18 и представляет различные задачи композиции, необходимые для исполнения составного мобильного приложения.

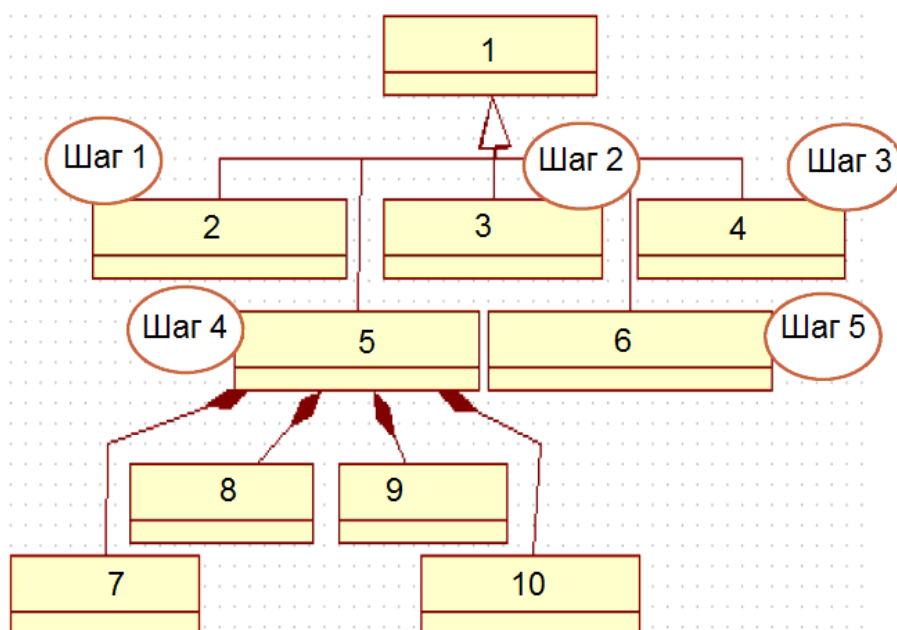


Рис. 4.18. Задачи процесса композиции: 1 – управление СМА; 2 – менеджер совместной работы; 3 – менеджер обнаружения; 4 – менеджер фильтрации; 5 – менеджер композиции; 6 – менеджер генерации СМА; 7 – задача адаптации; 8 – решатель; 9 – задача отбора; 10 – задача генерации профиля исполнения

С каждой ролью связывается определенный архитектурный элемент,

используемый для создания нужного мобильного приложения. Он основан на воплощении различных свойств и функциональных возможностей существующих композиционных механизмов.

4.2.2. Детализация процесса композиции

Шаг 1. Определение абстрактной функциональной архитектуры

Предлагаемый процесс направлен на то, чтобы в качестве первого шага абстрактно определить желаемое мобильное приложение. Таким образом, он должен в первую очередь сосредоточиться на определении различных функций, необходимых для создания будущего мобильного приложения, и обеспечить зависимости между этими желаемыми функциями. Эта задача решается менеджером совместной работы (рис. 4.18). Менеджер совместной работы определяет рабочий процесс и поток данных между идентифицированными функциями. Рабочий процесс планирует вызовы функций (т.е. устанавливает ссылки приоритета), а поток данных выражает обмен данными, входы и выходы, между ними (т.е. устанавливает ссылки использования). Помимо традиционных программных систем, разработка приложений на мобильных устройствах (например, смартфонах, планшетах и т.д.) ограничена их ограниченными ресурсами, такими как: небольшая память, вычислительная среда с питанием от батареи и доступность некоторых устройств (например, Wi-Fi, GPS, камера с автофокусом и т.д.) [4.54]. Для этой цели задача состоит не только в создании мобильных приложений с помощью композиции существующих программных объектов в соответствии с потребностями пользователей, но и в создании мобильных приложений, чувствительных к их контекстной информации (т.е. адаптированных к среде их исполнения). Таким образом, необходимо определить контекстную информацию среды развертывания. Она представляет все программные и аппаратные характеристики мобильного устройства, которые будут использоваться для

запуска нужного мобильного приложения. Эта контекстная информация описывается в трех категориях:

1) Контекст развертывания: представляет аппаратные характеристики мобильных устройств.

2) Контекст исполнения: представляет текущее состояние доступных устройств.

3) Платформа исполнения: представляет операционную систему мобильного устройства.

Эта различная контекстная информация может быть сформирована системой мобильного устройства или идентифицирована разработчиком.

Шаг 2. Поиск подходящих существующих программных объектов

После установки желаемого мобильного приложения на абстрактном уровне (т.е. в соответствии с архитектурным представлением, определенным на первом этапе процесса) необходимо соединить каждую абстрактную сущность, определенную в абстрактной функциональной архитектуре СМА, с их соответствующими существующими конкретными сущностями. Это - задача менеджера обнаружения, показанного на рис. 3. Менеджер обнаружения выполняет поиск и выбирает соответствующие конкретные объекты для каждой желаемой функции, загружая их через Интернет, где некоторые из них бесплатны, а другие платные (например интернет-магазины, такие как Google Play, App Store или магазин Windows Phone), или получая их из локальных расположений или с других устройств (например, ноутбуков, иных мобильных устройств и т.д.). Таким образом, менеджер обнаружения использует абстрактное функциональное представление нужного приложения для поиска различных конкретных объектов, которые могут быть лучше адаптированы к требованиям пользователя, описанным в этом представлении.

Результатом решения этой задачи обнаружения является набор подходящих конкретных сущностей для каждой абстрактной сущности.

Конкретные сущности, сопоставленные с одной и той же абстрактной сущностью, функционально эквивалентны, но могут различаться в нескольких нефункциональных аспектах. Для создания функционирующего мобильного приложения, адаптированного к мобильному устройству, которое будет его поддерживать, предлагается связать каждую сущность с определенным профилем исполнения (EP), который содержит все нефункциональные аспекты, представляющие необходимые условия для их исполнения (например необходимые устройства, такие как: Wi-Fi, GPS и т.д., размер, потребление энергии и т.д.).

Шаг 3. Выбор конкретных программных объектов с учетом контекста

3-й этап процесса гетерогенной композиции направлен на решение проблемы неоднородности мобильных устройств, которая может возникнуть при использовании такого мобильного устройства для развертывания желаемого СМА. Требования к качеству СМА представлены его адаптивностью с учетом контекста мобильного устройства, которое будет использоваться для его запуска. Таким образом, для обеспечения правильного развертывания и надлежащего функционирования составного мобильного приложения сначала необходимо убедиться, что их составные конкретные объекты могут быть адаптированы к текущему контексту мобильного устройства. Этот шаг соответствует операции фильтрации, целью которой является выбор среди всех конкретных сущностей каждой абстрактной сущности (т.е. результат шага 2) тех, которые лучше всего подходят в отношении контекста развертывания и исполнения мобильного устройства, на котором будет развернуто составное мобильное приложение. Эта возможность фильтрации возложена на менеджера фильтрации (рис. 4.18). Он основан на EP каждого конкретного объекта и различных характеристиках

мобильного устройства для исполнения сравнения, чтобы выбрать те, которые адаптированы к этим характеристикам.

Конкретная сущность соответствует контекстной информации мобильного устройства, если выполняются сравнения всех показателей исполнения со всеми характеристиками мобильного устройства. Таким образом, результатом этого шага является набор конкретных объектов, которые одновременно подходят для контекста развертывания и исполнения.

Шаг 4. Создание мобильного приложения

После соединения каждого абстрактного объекта с соответствующими конкретными объектами, зависящими от контекста, предлагаемый процесс на этом этапе предназначен для создания окончательной архитектуры приложения (т.е. абстрактной подробной архитектуры СМА) путем компоновки этих конкретных объектов в соответствии со связью, которая генерирует желаемое приложение с оптимальной стоимостью композиции.

Этот шаг направлен на создание мобильного приложения с более низкой стоимостью композиции путем выбора для каждого абстрактного объекта наиболее подходящего конкретного объекта среди всех соответствующих ему объектов в соответствии с минимальной стоимостью композиции для каждого из них. Стоимость композиции каждого конкретного объекта рассчитывается с точки зрения адаптивности на основе ограничений состава, описанных в этом разделе.

4.2.3. Гетерогенные и гомогенные композиции. Экзогенные и эндогенные посредники

Различные существующие подходы к композиции используют только один вид программных объектов для создания желаемого приложения. Как правило, они определяют приложение как набор

программных объектов одного и того же типа, используя, например, подход SOA [4.52] или компонентный подход [4.53] или другие парадигмы [4.51]. Основываясь на новой парадигме ХААС (что угодно как услуга) [4.55], преодолеем это ограничение, предоставляя описание, представляющее архитектуру СМА со службами, компонентами, приложениями отдельно или с разнородными объектами (т.е. с несколькими типами программных объектов). Таким образом, предлагаемый процесс позволит создавать мобильные приложения как экзогенную или эндогенную композицию, в которой каждая из составляющих их программных сущностей будет предоставлять определенную услугу. Однако проблема неоднородности в механизме композиции может представлять два вида проблем неоднородности:

- Неоднородная природа сущностей: составленные сущности не могут напрямую взаимодействовать, поскольку данные, которыми обмениваются эти сущности, обладают различными интерфейсами (например, микрофон обеспечивает аудиопоток, а музыкальный автомат должен вводить строки для исполнения своей задачи).

- Разнородный тип объектов: представляет собой координацию двух различных типов программных объектов (например, компонент, связанный с сервисом).

Предлагаемая метамодель направлена на решение этих проблем неоднородности, предлагая два вида посредников (рис. 4.17), среди которых:

- Эндогенный посредник, который преодолевает неоднородность между двумя объектами различной природы, обмен данными может потребовать некоторой трансформации, чтобы быть понятным.

Эндогенный посредник представляет посреднические услуги, которые выбираются для обеспечения этих преобразований данных.

- Экзогенный посредник предназначен для устранения

неоднородности между двумя субъектами разного рода. Эти сущности не могут напрямую общаться из-за их разного типа. Экзогенный посредник стремится инкапсулировать связанные разнородные сущности, он создает хорошо сформированный интерфейс для каждой из них, чтобы воспользоваться их услугами, но только с манипулированием необходимыми входными и выходными данными независимо от языковых реализаций этих составляющих сущностей.

Для этой цели этот процесс композиции обрабатывает четыре типа композиции, как показано в табл. 4.2.

Таблица 4.2

Типы композиций

Тип композиции	Проблемы гетерогенности	Предлагаемые посредники
Гетерогенная экзогенная композиция	Гетерогенная природа сущностей Гетерогенный тип сущностей	Эндогенные посредники Экзогенные посредники
Гомогенная экзогенная композиция	Гетерогенный тип сущностей	Экзогенные посредники
Гетерогенная эндогенная композиция	Гетерогенная природа сущностей	Эндогенные посредники
Гомогенная эндогенная композиция	Отсутствие проблем гетерогенности	Посредники не требуются

Менеджер композиций стремится обрабатывать абстрактную функциональную архитектуру СМА, чтобы получить абстрактную подробную архитектуру СМА, заменяя каждую функциональность соответствующей конкретной сущностью с учетом контекста, которая

может реализовать ее в отношении стоимости ее композиции.

Менеджеру композиций необходимо выполнить четыре задачи для достижения своей цели:

1. Задача посредничества, представляющая гетерогенную и/или экзогенную композицию, управляет ссылками использования, которые представляют поток данных между желаемыми функциями, и определяет, необходимо ли связать эту связь с посредничеством служб или/и инкапсулировать составленные объекты.

2. Основываясь на ограничениях композиции объектов, менеджер композиций выполняет задачу выбора, рассчитывая стоимость композиции для каждого конкретного объекта, чтобы выбрать наиболее подходящий, который позволит создать желаемое мобильное приложение с более низкой стоимостью композиции. Состав адаптируемых конкретных объектов недостаточен для обеспечения того, чтобы само составное приложение было адаптировано к контексту мобильного устройства.

3. Менеджер композиций должен проверять после каждого шага композиции, достаточна ли фактическая свободная емкость хранилища для развертывания составного приложения на этом мобильном устройстве (задача адаптации). Или, чтобы потребляемая энергия для обработки приложения не превышала текущий уровень заряда батареи. Если одно из этих двух ограничений не выполняется, диспетчер композиций запускает перекомпозицию приложения с использованием альтернативных конкретных объектов, выбранных на шаге 3.

4. И последнее, но не менее важное: необходимо создать для каждого составленного мобильного приложения свой собственный ЕР и определить его собственные свойства. Менеджер композиций обладает потенциалом для исполнения ЕР СМА в соответствии с различными характеристиками их компонентов (задача создания ЕР).

Шаг 5. Создание исполняемого приложения

Мобильные приложения составляются визуально в соответствии с предлагаемым архитектурным представлением без необходимости написания каких-либо строк кода. Исполняемая модель будет сгенерирована из описания СМА после поиска, фильтрации и выбора наиболее подходящих конкретных объектов (т.е. абстрактной подробной архитектуры СМА), как показано на рис. 4.18. Таким образом, предлагаемый подход справляется с неоднородностью мобильных платформ. Он способен поддерживать разнородные целевые среды, начиная от архитектуры СМА и заканчивая мобильными платформами. Исходя из этой архитектурной модели, генерация конкретного СМА (т.е. кода приложения) для конкретной платформы основана на механизмах управляемых моделью разработки с использованием преобразований модель в код. Эта задача входит в обязанности менеджера СМА.

Этот шаг доказывает, что предложенный процесс обладает потенциалом для развертывания и миграции одного и того же СМА между различными мобильными платформами (например, Android, iOS).

4.3. Проектирование механизма межагентного взаимодействия

В рамках требований к системе можно выделить следующие прецеденты:

- регистрация сервиса;
- удаление сервиса;
- поиск сервиса по наименованию;
- чтение сервиса;
- вызов сервиса;
- обновление сервиса;

— аутентификация пользователя.

Диаграмма прецедентов для разрабатываемой системы представлена на рис. 4.19.

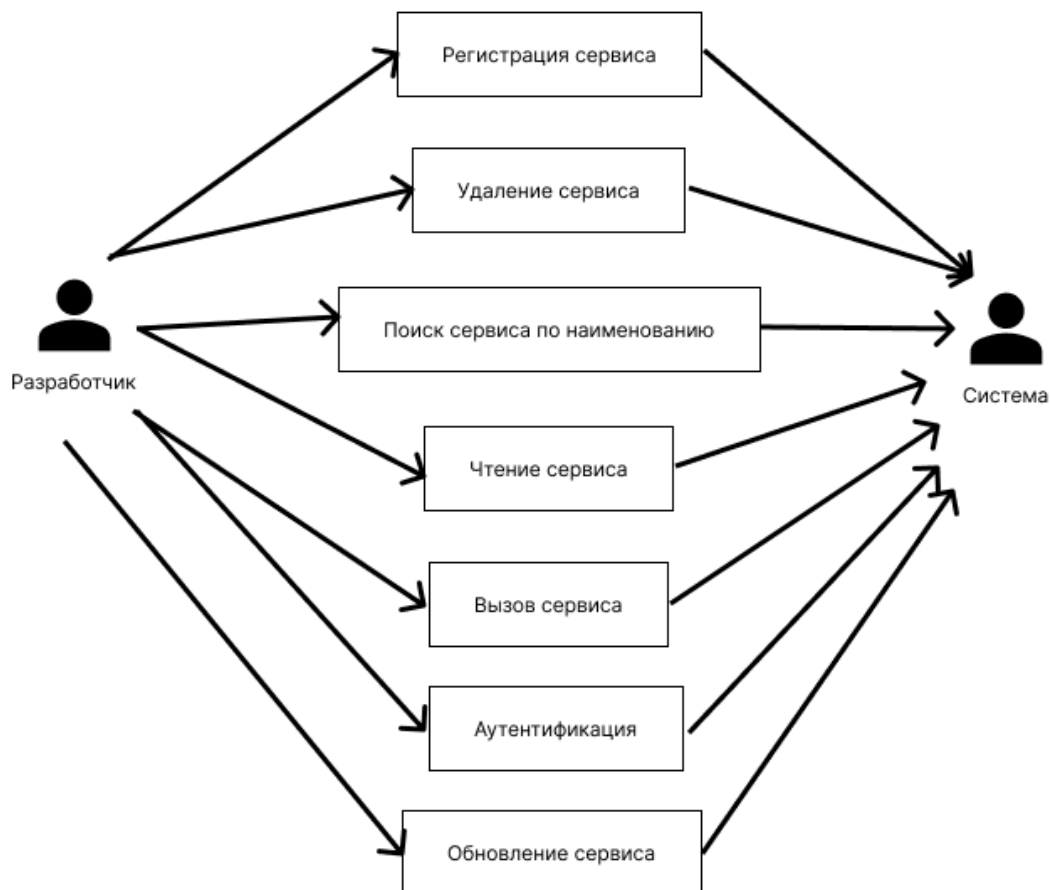


Рис. 4.19. Диаграмма прецедентов типового интерфейса

Диаграмма классов для разрабатываемой системы представлена на рис. 4.20. В качестве примера на рис. 4.21 представлена диаграмма деятельности для прецедента «Обновление сервиса».

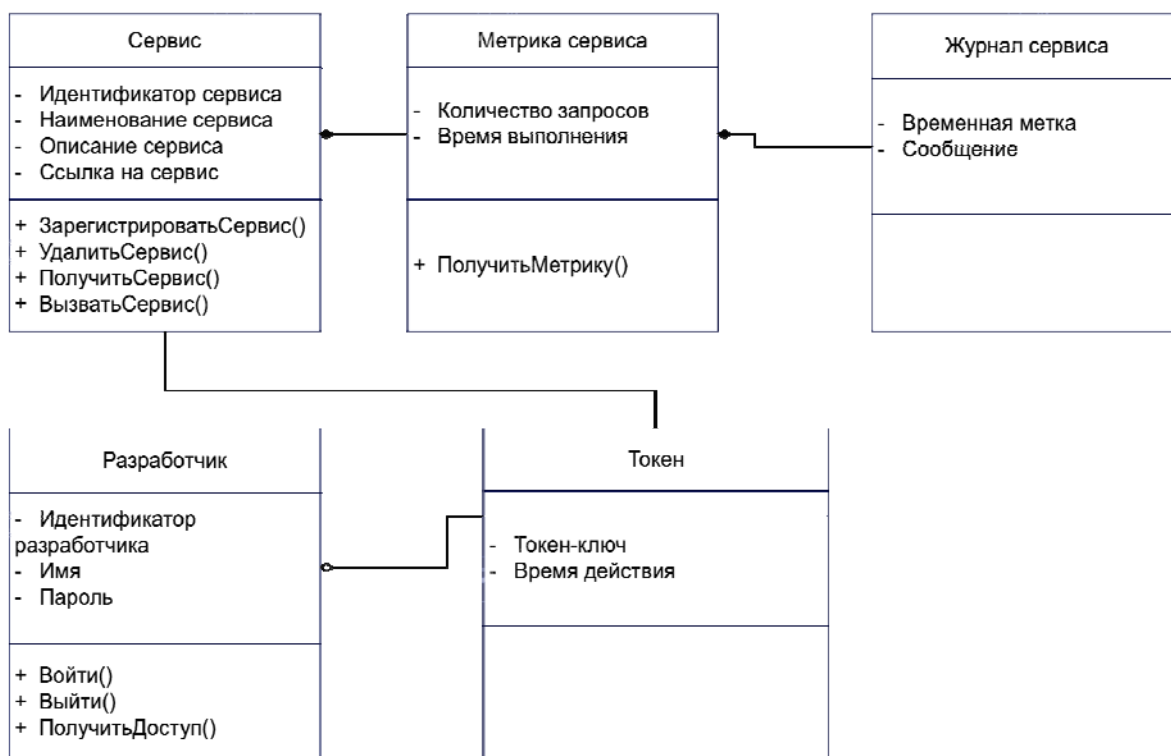


Рис. 4.20. Диаграмма классов для разрабатываемого интерфейса

Основные результаты внедрены в Научно-исследовательском институте вычислительных комплексов им. М. А. Карцева» (г. Москва) при проектировании инфокоммуникационных систем на основе мобильных приложений, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

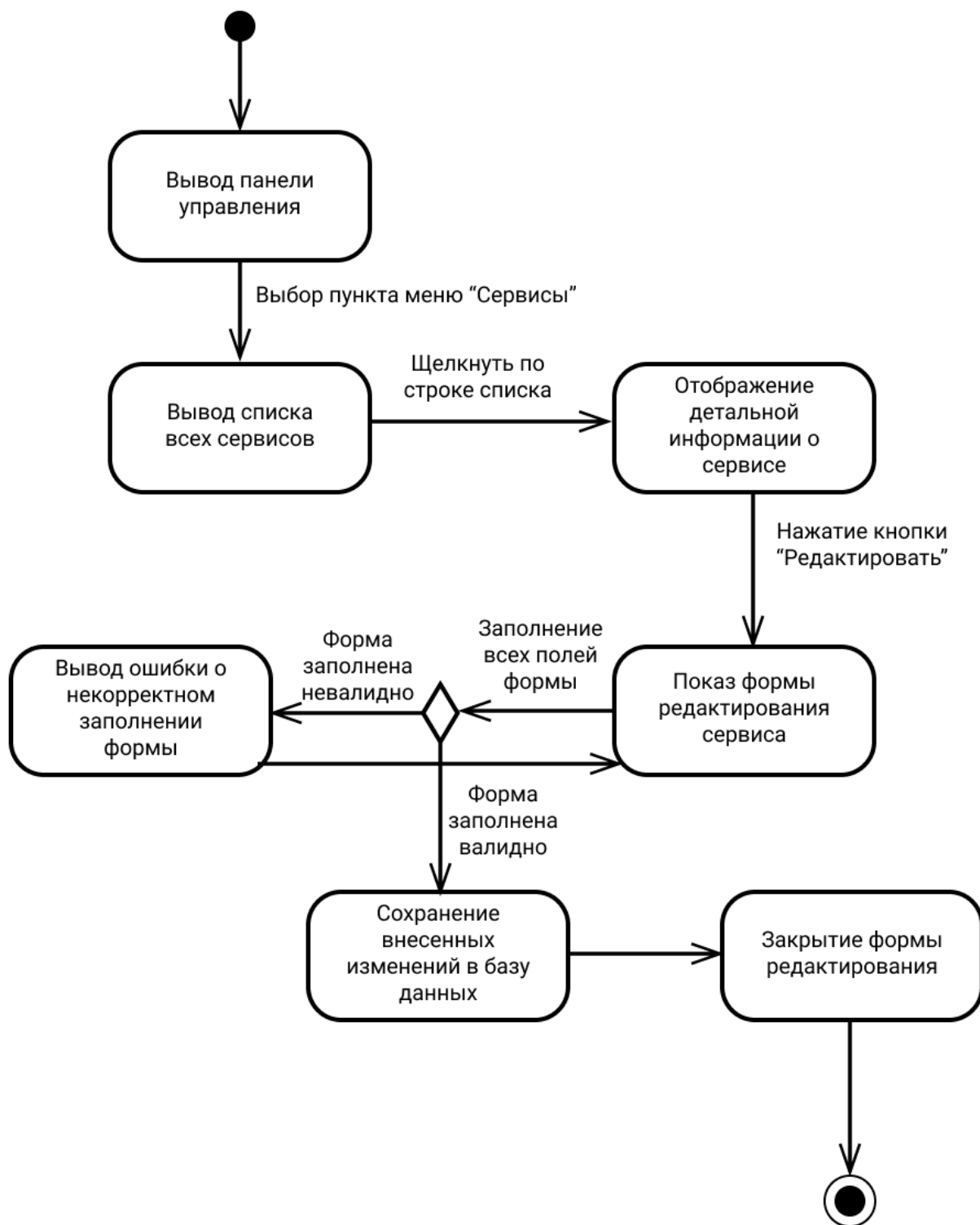


Рис. 4.21. Диаграмма деятельности для прецедента «Обновление сервиса»

4.4. Выводы

1. Увеличение времени жизни сети (NL) является важным требованием в беспроводных сенсорных сетях (WSN). Одним из методов увеличения NL является использование деревьев агрегации данных (DAT). DAT улучшают NL, комбинируя преимущества энергоэффективности как агрегации данных (DA), так и маршрутизации на основе дерева. Пока для построения DAT широко используются как централизованные, так и распределенные стратегии, мы предлагаем комбинированный подход для улучшения NL. Этот подход снижает накладные расходы на связь и ослабляет потребность в полной информации о сети на приемнике. В этой работе такой комбинированный подход называется подходом расширенного локального просмотра (ELV). На основе ELV предложены два алгоритма построения DAT: ELV с фиксированным приемником (ELVF) и ELV со случайным приемником (ELVR). Оба алгоритма используют основанный на эвристике метод восстановления локального пути (LPR) и жадный алгоритм расширенного восстановления пути (EPR). Используя эти методы планируется последовательность DAT, которые в совокупности улучшают NL, а также уменьшают накладные расходы на реструктуризацию DAT. Эффективность ELVF и ELVR оценивается с помощью строгих экспериментов, и результаты моделирования показывают, что предложенные алгоритмы улучшают NL и масштабируются при различных значениях коэффициента DA.

Алгоритм ELVF использует основанный на эвристике алгоритм LPR и жадный алгоритм EPR. Эти алгоритмы идентифицируют последовательность DAT, которые в совокупности увеличивают NL. Результаты моделирования показали, что ELVF является масштабируемым и увеличивает NL на 6-47% по сравнению с современными алгоритмами.

Алгоритм ELVR случайным образом выбирает расположение

приемника и снижает вероятность совершения атаки на него. Результаты моделирования для ELVR показывают, что NL увеличивается на 80% по сравнению с ELVF, и демонстрируют его расширяемость.

Хотя создание DAT и увеличивает NL, что продемонстрировано результатами моделирования, оно также требует дополнительных затрат на реструктуризацию DAT. Анализ длины DAT с помощью строгого моделирования показывает, что ELVF и ELVR определяют более короткие графы, которые подходят для агрессивных и критических сред WSN, где вмешательство человека невозможно.

Таким образом:

- Предложены два алгоритма построения DAT на основе ELV: ELV с фиксированным приемником (ELVF) и ELV со случайным приемником (ELVR) для задачи DATRL. И ELVF, и ELVR составляют последовательность DAT с использованием реструктуризации дерева и улучшают NL.

- В алгоритме ELVF положение приемника фиксировано. ELVF использует основанный на эвристике метод восстановления локального пути (LPR) и жадный алгоритм расширенного восстановления пути (EPR) для улучшения NL.

- В алгоритме ELVR приемник размещается в случайно выбранной позиции, которая подходит для агрессивных сред и снижает вероятность атаки на приемник.

- Оценивается производительность предложенных алгоритмов, и результаты экспериментов показывают, что алгоритмы улучшают NL и демонстрируют масштабируемость. Кроме того, результаты демонстрируют, что ELVF и ELVR снижают затраты на реструктуризацию DAT.

2. Представлена концептуальная основа для предложенной идеи обеспечения процесса гетерогенной композиции для мобильных

приложений. Этот процесс определяется на архитектурном уровне. Он направлен на удовлетворение потребностей пользователей и создание мобильных приложений, которые чувствительны и адаптированы к контекстной информации мобильного устройства, на котором они будут установлены, а также для облегчения задачи создания мобильных приложений при обеспечении эффективности созданного приложения.

3. Основные результаты внедрены в Научно-исследовательском институте вычислительных комплексов им. М. А. Карцева» (г. Москва) при проектировании инфокоммуникационных систем на основе мобильных приложений, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Литература к главе 4

4.1. Madden S, Franklin MJ, Hellerstein JM, Hong W. TAG: a tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper Syst Rev. 2002;36(SI):131-146. <https://doi.org/10.1145/844128.844142>.

4.2. Fasolo E, Rossi M, Widmer J, Zorzi M. In-network aggregation techniques for wireless sensor networks: a survey. Wireless Commun. 2007;14(2):70-87. <https://doi.org/10.1109/MWC.2007.358967>.

4.3. Rajagopalan R, Varshney PK. Data aggregation techniques in sensor networks: a survey. IEEE Commun Surv Tut. 2006;8:48-63.

4.4. Liu C, Cao G. Distributed monitoring and aggregation in wireless sensor networks. In: INFOCOM, 2010 Proceedings IEEE; 2010:1-9.

4.5. Kuo T-W, Lin KC-J, Tsai M-J. On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: Npcompleteness and approximation algorithms. IEEE Trans Comput. 2016;65(10):3109-3121.

4.6. Nguyen N-T, Liu B-H, Pham V-T, Luo Y-S. On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data aggregation trees. Comput Netw. 2016; 105(C): 99-110. <https://doi.org/10.1016/j.comnet.2016.05.022>.

4.7. Wu Y, Mao Z, Fahmy S, Shroff NB. Constructing maximum-lifetime data-gathering forests in sensor networks. IEEE/ACM Trans Netw. 2010;18(5):1571-1584.

4.8. Buragohain C, Agrawal D, Suri S. Power aware routing for sensor databases. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol. 3 IEEE; 2005:1747-1757.

4.9. Tan HÖ, Körpeolu I. Power efficient data gathering and aggregation in wireless sensor networks. ACM Sigmod Record. 2003;32(4):66-71.

4.10. Shan M, Chen G, Luo D, Zhu X, Wu X. Building maximum lifetime shortest path data aggregation trees in wireless sensor networks. *ACM Trans Sensor Netw (TOSN)*. 2014;11(1):11.

4.11. He J, Ji S, Pan Y, Li Y. Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks. *IEEE Trans Parallel Distrib Syst*. 2014;25(7):1681-1690.

4.12. Ding M, Cheng X, Xue G. Aggregation tree construction in sensor networks. In: *Vehicular Technology Conference, 2003. vtc 2003-fall. 2003 IEEE 58th, Vol. 4 IEEE*; 2003:2168-2172.

4.13. Lee M, Wong VW. An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks. In: *Communications, Computers and Signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on IEEE*; 2005:300-303.

4.14. Dasgupta K, Kalpakis K, Namjoshi P. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In: *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, Vol. 3 IEEE*; 2003:1948-1953.

4.15. Xue Y, Cui Y, Nahrstedt K. Maximizing lifetime for data aggregation in wireless sensor networks. *Mobile Netw Appl*. 2005;10(6):853-864.

4.16. Lin H-C, Li F-J, Wang K-Y. Constructing maximum-lifetime data gathering trees in sensor networks with data aggregation. In: *Communications (ICC), 2010 IEEE International Conference on IEEE*; 2010:1-6.

4.17. Matsuura H. Maximizing lifetime of multiple data aggregation trees in wireless sensor networks. In: *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP IEEE*; 2016:605-611.

4.18. Xie S, Wang Y. Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wirel Personal Commun*. 2014;78(1):231-246. <https://doi.org/10.1007/s11277-014-1748-5>.

4.19. Yildiz HU. Investigation of maximum lifetime and minimum delay trade-off in underwater sensor networks. *Int J Commun Syst.* 2019; 32(7):e3924. <https://doi.org/10.1002/dac.3924>

4.20. Ahmed YE, Adjallah KH, Stock R, Kacem I, Babiker SF. NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring. *Comput Indust Eng.* 2018;115:17-25.

<http://www.sciencedirect.com/science/article/pii/S0360835217304667>.

4.21. Al-Kiyumi RM, Foh CH, Vural S, Chatzimisios P, Tafazolli R. Fuzzy logic-based routing algorithm for lifetime enhancement in heterogeneous wireless sensor networks. *IEEE Trans Green Commun Netw.* 2018;2(2):517-532.

4.22. Liu X. An optimal-distance-based transmission strategy for lifetime maximization of wireless sensor networks. *IEEE Sens J.* 2015;15(6): 3484-3491.

4.23. Kale PA, Nene MJ. Path reestablishment in wireless sensor networks. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET); 2017:1659-1663.

4.24. Sadeghi F, Avokh A. Load-balanced data gathering in internet of things using an energy-aware cuckoo-search algorithm. *International Journal of Communication Systems.* 2020;33(9):e4385. <https://doi.org/10.1002/dac.4385>.

4.25. Liang W, Xu Y, Shi J, Luo J. Aggregate node placement for maximizing network lifetime in sensor networks. *Wirel Commun Mobile Comput.* 2012;12(3):219-235. <https://doi.org/10.1002/wcm.952>.

4.26. Jothiprakasam S, Muthial C. A lifetime enhancement protocol in cluster-based wireless sensor networks with guaranteed delay. *Int J Commun Syst.* 2019;32(12):e3983. <https://doi.org/10.1002/dac.3983>.

4.27. Kale PA, Nene MJ. Scheduling of data aggregation trees using local heuristics to enhance network lifetime in sensor networks. *Computer Networks.* 2019;160:51-64.

4.28. Kale PA, Nene MJ. Data aggregation trees with QoS in sensor networks. In: 2019 IEEE 5th International Conference for Convergence in Technology (i2ct); 2019:1-5.

4.29. Liu X, Liu A, Wang T, Ota K, Dong M, Liu Y, Cai Z. Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks. *J Parall Distrib Comput*. 2020;135:140-155.

4.30. Liu X, Zhao M, Liu A, Wong KKL. Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks. *Inf Fusion*. 2020;53:183-195.

4.31. Wang J, Gao Y, Yin X, Li F, Kim H-J. An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. *Wirel Commun Mobile Comput*. 2018;2018.

4.32. Wang J, Gao Y, Liu W, Wu W, Lim S-J. An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. *Comput Mater Contin*. 2019;58:711-725.

4.33. Wang J, Gu X, Liu W, Sangaiah AK, Kim H-J. An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Human-centric Comput Inf Sci*. 2019;9(1):1-14.

4.34. Wang J, Gao Y, Wang K, Sangaiah AK, Lim S-J. An affinity propagation-based self-adaptive clustering method for wireless sensor networks. *Sensors*. 2019;19(11):2579.

4.35. Wang J, Gao Y, Zhou C, Sherratt RS, Wang L. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns. *Comput Mater Continua*. 2020; 62(2):695-711. <http://www.techscience.com/cmc/v62n2/38271>.

4.36. Zhou F, Chen Z, Guo S, Li J. Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs. *IEEE Sens J*. 2016; 16(22):8167-8177.

4.37. Kale PA, Nene MJ. Quality aware data aggregation trees in sensor

networks. In: Intelligent Communication Technologies and Virtual Mobile Networks Springer; 2019:557-567.

4.38. Imon SKA, Khan A, Di Francesco M, Das SK. Energy-efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks. IEEE/ACM Trans Netw (TON). 2015;23(5):1401-1415.

4.39. Kale PA, Nene MJ. Asymptotic cost estimation for scheduling data aggregation trees in sensor networks. In: 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS); 2019:409-414.

4.40. Lu Y, Comsa I-S, Kuonen P, Hirsbrunner B. Adaptive data aggregation with probabilistic routing in wireless sensor networks. Wirel Netw. 2016;22(8):2485-2499.

4.41. Горшков А.В., Кравец О.Я. Расширение функций мобильных программных агентов при решении задач балансировки нагрузки в распределенных системах управления// Системы управления и информационные технологии, №3(89), 2022. – С. 78-85.

4.42. Горшков А.В., Кравец О.Я. Экспериментальное исследование эффективности расширения функций мобильных программных агентов при решении задач балансировки нагрузки// Информационные технологии моделирования и управления, №3(129), 2022. – С. 220-231.

4.43. Горшков А.В., Кравец О.Я. Метамоделирование композиционного процесса разработки составных мобильных приложений// Прогрессивные технологии и процессы: Сб. науч. статей 9-й Всеросс. НТК с Междунар. участием. - Курск: Юго-Зап. гос. ун-т, 2022. – С. 64-67.

4.44. Панченко Н.В. Разработка мобильных приложений// Тенденции развития науки и образования, 2019, 12, с. 26-29.

4.45. Hock-Koon A., Oussalah M. Expliciting a composite service by a metamodeling approach// Research Challenges in Information Science (RCIS), 2010 Fourth IEEE Int. Conf. on. 2010.

4.46. Chakraborty D. et al. Service composition for mobile environments// Mobile Networks and Applications, 2005, 10(4), p. 435-451.

4.47. Wu Z. et al. Automatic composition of semantic web services using process and data mediation// ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume SAIC, Funchal, Madeira, Portugal, 2007.

4.48. Hock-Koon A., Oussalah M. Composite service metamodel and auto composition// Journal of Computational Methods in Science and Engineering, 2010, 10, p. 215-229.

4.49. Rosa R.E.V., Lucena V.F. Jr. Smart composition of reusable software components in mobile application product lines// Proc. of the 2nd Int. ACM Workshop on Product Line Approaches in Software Engineering. 2011.

4.50. Furno A., Zimeo E. Context-aware Composition of Semantic Web Services// Mobile Networks and Applications, 2014. 19(2), p. 235-248.

4.51. Abdelkrim A., Hock-Koon A., and Oussalah M.C. Chapter 1: Object-Oriented, Component-Based, Agent-Oriented and Service Oriented Paradigms, // Software Architectures 1, Wiley-ISTE, 2014.

4.52. Sward R., Boleng J. Service-oriented architecture (SOA) concepts and implementations// ACM SIGAda Ada Letters, 2011, 31(3). DOI: 10.1145/2070336.2070340.

4.53. Jifeng H., Li X., Liu Z., Component-based software engineering// Theoretical Aspects of Computing–ICTAC, Springer, 2005, p. 70-95.

4.54. Zhang X. et al. Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing// Mobile Networks and Applications, 2011. 16(3), p. 270-284.

4.55. Gao F., Curry E., Ali M.I., Mileo A. QoS-Aware Complex Event Service Composition and Optimization Using Genetic Algorithms// Lecture Notes in Computer Science, 2014, vol. 8831. DOI: 10.1007/978-3-662-45391-9_28.

Заключение

Целью работы являлась разработка математического и программного обеспечения управления инфокоммуникационными службами на основе на основе систем мобильных агентов.

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Создана марковская модель распространения информации мобильными агентами в звездообразной сети, обеспечивающая уменьшение вычислительной сложности расчета ожидаемого времени распространения с $O(N^3)$ до $O(N^2)$.

2. Разработаны алгоритмы балансировки нагрузки, обеспечивающие улучшение равномерности распределения нагрузки.

3. Предложена архитектура системы мобильных агентов, обеспечивающая динамическое перемещение служб для повышения доступности услуг.

4. Разработан алгоритм расширенного локального просмотра беспроводной агентной сети с фиксированным или случайным приемником данных, обеспечивающий снижение накладных расходов на взаимодействие и требований к полноте информации о сети на приемнике.

5. Предложена архитектура составных мобильных приложений, обеспечивающая реализацию процесса гетерогенной композиции мобильного приложения с учетом требований заказчика.

6. Элементы программного обеспечения зарегистрированы в ФИПС.

Список использованных источников

1. Гамильтон С. Управление цепочками поставок с Microsoft Axapta. – М.: Альпина Бизнес Букс, 2005, 352 с.
2. Ги К. Введение в локально-вычислительные сети. – М.: Радио и связь, 2000. – 190 с.
3. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания, 2-е изд., М.:Наука, 1987. - 336 С.
4. Говорский А.Э., Кравец О.Я. Особенности взаимодействия подсистем при решении задач интегрального обслуживания неоднородного трафика // Системы управления и информационные технологии. 2008. Т. 31. № 1.1. С. 141-146.
5. Горшков А.В. Исследование механизма распространения информации в мультиагентной системе на основе марковских процессов// Системы управления и информационные технологии, №4(90), 2022. – С. 42-48.
6. Горшков А.В. Проблема миграции услуг на основе мобильных агентов// Достижения науки и технологий-ДНиТ-11-2023. Сборник научных статей по материалам II Всероссийской научной конференции. - Красноярск, 2023. С. 340-344.
7. Горшков А.В., Кравец О.Я. Архитектура композиционного процесса разработки мобильных приложений на основе гетерогенных программных объектов// Прогрессивные технологии и процессы: Сб. науч. статей 9-й Всеросс. НТК с Междунар. участием. - Курск: Юго-Зап. гос. ун-т, 2022. – С. 68-71.
8. Горшков А.В., Кравец О.Я. Исследование механизма распространения информации в мультиагентной системе во временном окне// Моделирование, оптимизация и информационные технологии. 2023;

2023;11(1). <https://moitvivi.ru/ru/journal/pdf?id=1323>. DOI: 10.26102/2310-6018/2023.40.1.023

9. Горшков А.В., Кравец О.Я. Метамоделирование композиционного процесса разработки составных мобильных приложений// Прогрессивные технологии и процессы: Сб. науч. статей 9-й Всеросс. НТК с Междунар. участием. - Курск: Юго-Зап. гос. ун-т, 2022. – С. 64-67.

10. Горшков А.В., Кравец О.Я. Применение мобильных агентов для балансировки нагрузки в распределенных системах управления// Решение: матер. 11-й Всеросс. НПК. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2022. – С. 196-199.

11. Горшков А.В., Кравец О.Я. Расширение функций мобильных программных агентов при решении задач балансировки нагрузки в распределенных системах управления// Системы управления и информационные технологии, №3(89), 2022. – С. 78-85.

12. Горшков А.В., Кравец О.Я. Экспериментальное исследование эффективности расширения функций мобильных программных агентов при решении задач балансировки нагрузки// Информационные технологии моделирования и управления, №3(129), 2022. – С. 220-231.

13. Горшков А.В., Кравец О.Я., Аль Мусави О.А.Р., Гребенникова Н.И. Программа визуального моделирования миграции агентов в мультиагентной системе. – Свидетельство о регистрации программы для ЭВМ № 2022685251 от 22.12.2022. – М.: ФИПС, 2022.

14. Гриневич А.И., Кулямин В.В., Марковцев Д.А., Петренко А.К., Рубанов В.В., Хорошилов А.В. Использование формальных методов для обеспечения соблюдения программных стандартов. – Тр. института системного программирования РАН. - 2006. - Т.10, с. 51-68.

15. Грязнов Н.Г., Димитриев Ю.К., Мелентьев В.А. Оптимизация отказоустойчивого вложения диагностического графа в тороидальные

структуры живучих вычислительных систем// Автоматика и телемеханика. 2003. № 4. С. 133-152.

16. Дастин Э., Рэшка Д., Джон П. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. - М.: Лори, 2003.

17. Дейтел Х., Дейтел П., Нието Т. Как программировать для Internet & WWW М.: Бином, 2002, - 1184 С.

18. Денисов А.А., Колесников Д.Н. Теория больших систем управления. Л.: Энергоиздат, 1982. - 288 С.

19. Диго С.М. Проектирование и использование баз данных. М.: Финансы и статистика, 1995. -208 С.

20. Жожикашвили В.А., Вишневский В.М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. М.: Радио и связь, 1988. – 192 С.

21. Журков А.П., Аминев Д.А., Гусева П.А., Мирошниченко С.С., Петросян П.А. Анализ возможностей применения подходов самодиагностирования к распределенной радиотехнической системе наблюдения// Системы управления, связи и безопасности. 2015. № 4. С. 114-122.

22. Задорожный В.Н. Модели и системы. Анализ научного мышления. - Омск, ОмГТУ, 1999. - 100 с.

23. Задорожный В.Н. Статистическое моделирование. - Омск: Изд-во ОмГТУ, 1996. - 92с.

24. Иванов П. Управление информационными системами: базовые концепции и тенденции развития // Открытые системы, №4, 1999, С. 37-43.

25. Камер Д. Компьютерные сети и Internet М.: Вильямс, 2002.- 640 С.

26. Кастаньетто Д. и др. Профессиональное PHP программирование. М.: Символ – Плюс, 2001. - 912 С.

27. Кениг Д., Штоян Д. Методы теории массового обслуживания.- М.: Радио и связь, 1981. – 127 с.
28. Киллелиа П. Тюнинг WEB-сервера. СПб.: Питер, 2003. - 528 С.
29. Клейнен Дж. Статистические методы в имитационном моделировании. - М.: Статистика, 1978. - Вып. 1. - 221 с.; Вып. 2 - 335 с.
30. Клейнрок Л. Вычислительные системы с очередями: Пер. с англ. М.: Мир, 1979 - 600 С.
31. Клейнрок Л. Теория массового обслуживания: Пер. с англ. М.: Машиностроение, 1979 – 432 С.
32. Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финстат, 2002. - 800 С.
33. Кокс Д. Р., Смит У. Л. Теория очередей: Пер. с англ. М.: Мир, 1966 - 218 С.
34. Колесов Ю.Б., Сениченков Ю.Б., Визуальное моделирование, СПб.: Мир и Семья, 2000. - 256 С.
35. Конвей Р., Максвелл В., Миллер Л. Теория расписаний: Пер с англ. / Под ред. Г.П. Башарина. – М.: Наука, 1975 – 159 с.
36. Ланг К., Чоу Д. Публикация баз данных в Интернете. – СПб.: Символ-Плюс, 2004.
37. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети. Л.: Энергоатомиздат, 1987. - 256 С.
38. Лебедев А.Н., Чернявский Е.А. Вероятностные методы в вычислительной технике. М.: Высшая школа, 1986. - 312 С.
39. Левенчук А. Интранет предлагает решения для корпорации// Рынок ценных бумаг, №16, 1996.
40. Липский Н. Комбинаторика для программистов. М.: Мир, 1985. – 374 С.
41. Мазуркевич А., Еловой Д. PHP: настольная книга программиста. М.: Новое знание 2003. -480 С.

42. Майерс Г. Надежность программного обеспечения. - М.: Мир, 1980.
43. Мальцева С.В. Информационное моделирование WEB-ресурсов Интернет. М.: Глобус, 2003. - 216 С.
44. Матвеев В.Ф., Ушаков В.Г. Системы массового обслуживания. М.: Изд-во МГУ, 1984. – 240 С.
45. Мельтцер К., Михальски Б. Разработка CGI-приложений на Perl. М.: Вильямс, 2001. - 400 С.
46. Мещеряков Е.В., Хомоненко А.Д. Публикация баз данных в Интернете Спб.: BHV-Санкт-Петербург, 2001. - 560 С.
47. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 2-изд. СПб: Питер-пресс, 2002. - 864 С.
48. Павловский Ю. Н. Имитационные модели и системы. М.: ФАЗИС, 2000. - 144 С.
49. Панченко Н.В. Разработка мобильных приложений// Тенденции развития науки и образования, 2019, 12, с. 26-29.
50. Перегудов Ф.П., Тарасенко Ф.П. Введение в системный анализ. М: Высшая школа, 1989. - 367 С.
51. Петренко А., Бритвина Е., Грошев С., Монахов А., Петренко О. Тестирование сетевой инфраструктуры// Открытые системы, 09/2003.
52. Профессиональные стандарты в области информационных технологий. - М.: АП КИТ, 2008. - 616 с.
53. Ройс У. Управление проектами по созданию программного обеспечения. - М.: Лори, 2002.
54. Свами М., Тхуласираман К. Графы, сети и алгоритмы: Пер. с англ. М.: Мир, 1984. - 455 С.
55. Седова Н.А. Формирование лингвистических переменных для задач судовождения// Эксплуатация морского транспорта. – 2013. – №2(72). – С. 19-23.

56. Седова Н.А., Перечёсов В.С., Седов В.А. Удержание судна на курсе на базе нечеткой логики с учетом скорости судна// Автоматизация процессов управления. – 2013. – № 2. – С.74-79.

57. Советов Б.Я., Яковлев С.А. Моделирование систем //3-е изд., М:Высшая школа, 2001. - 344 С.

58. Создание Intranet. Официальное руководство Microsoft. - СПб.: BHV, 1998.

59. Спицын А.А. Алгоритм настройки параметров расписания облачных вычислений на основе оптимизации роя частиц// Вестник Рязанского государственного радиотехнического университета, № 75, 2021. С. 44-52.

60. Спицын А.А. Исследование расширенной архитектурной модели облачных вычислений // Сб. ст. Междунар. НПК «Материалы и методы инновационных научно-практических исследований и разработок» (Пенза, 17.03.2021 г.). – Уфа: OMEGA SCIENCE, 2021. – с. 36-44.

61. Спицын А.А. Оптимизация планирования в облачных вычислениях// IV Всероссийская научная конференция с международным участием «Информационные технологии в моделировании и управлении: подходы, методы, решения». – Тольятти, 2021. С. 315-323.

62. Спицын А.А. Оптимизированный по стоимости эвристический алгоритм для планирования рабочих процессов в облачной среде сервисов IaaS// Системы управления и информационные технологии, №1(83), 2021. – С. 30-37.

63. Спицын А.А. Результаты вычислительного эксперимента по планированию облачных вычислений на основе оптимизации роя частиц// Информационные технологии моделирования и управления, №1(123), 2021. – С. 68-79.

64. Спицын А.А. Управление расписанием облачных вычислений на

основе оптимизации роя частиц// Решение: матер. 11-й Всеросс. НПК. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2022. – С. 255-259.

65. Спицын А.А., Львович И.Я., Зеленина А.Н. Диспетчерское управление распределением процессов в облачных вычислениях. – Свидетельство о регистрации программы для ЭВМ № 2021611994 от 10.02.2021. М.: ФИПС, 2021.

66. Спицын А.А., Мутин Д.И. Оптимизация миграции виртуальных машин в облачных вычислениях с помощью эффективного алгоритма размещения// Матер. 16-й Международной научно-технической конференции «Современные инструментальные системы, информационные технологии и инновации» (МТО-56). – Курск, 2021. С. 251-256.

67. Спицын А.А., Мутин Д.И. Распределение ресурсов и планирование заданий в облачной среде на основе алгоритма оптимизации роя частиц и R-фактора// Моделирование, оптимизация и информационные технологии. 2020; 8(4). <https://moitvivi.ru/ru/journal/pdf?id=869>. DOI: 10.26102/2310-6018/2020.31.4.023.

68. Стрелкова Е. Интеграция данных предприятия / Открытые системы, №4, 2003, С. 58-60.

69. Технология системного моделирования/ Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов и др. / Под ред. С.В. Емельянова. – М.: Машиностроение, 1988. – 320 с.

70. Томсетт Р. Радикальное управление ИТ проектами. М.: Лори, 2005.

71. Увайсов С.У., Иванов И.А., Кошелев Н.А. Методика обеспечения диагностируемости электронных средств космических аппаратов по ранговому критерию на ранних этапах проектирования// Качество. Инновации. Образование. 2012. № 1 (80). С. 60-63.

72. Уемов А.И. Системный подход и общая теория систем. М.: Мысль, 1978.- 272 С.
73. Управление качеством продукции. Введение в системы менеджмента качества / С. Пономарев, С. Мищенко, В.Я. Белобрагин. М.: Стандарты и качество, 2005.
74. Феллер В. Введение в теорию вероятностей и её приложения: в 2-х т. - М.: Мир, 1967. - т.1, 498 с.
75. Феррари Д. Оценка производительности вычислительных систем. М.: Мир, 1981. - 576 С.
76. Французов Д. Оценка производительности вычислительных систем. Открытые системы, №2, 1996, С. 58-66.
77. Фролов А.В. Фролов Г.В. Локальные сети персональных компьютеров. М.: ДИАЛОГ-МИФИ, 1993. - 176 С.
78. Харари Ф. Теория графов. М.: Мир, 1973. - 300 С.
79. Харрингтон Дж. Проектирование реляционных баз данных. М: Лори-Пресс, 2000. - 230 С.
80. Харт Д.М. Системное программирование в среде Win32 - 2-е издание. М.: Вильямс, 2001. 464 С.
81. Хилайер С., Мизик Д. Программирование Active Server Pages. М.: Русская редакция. 2000. - 320 С.
82. Храмцов П.Б., Брик С.А. и др. Основы web-технологий. М.: ИНТУИТ.ру, 2003. – 512 С.
83. Цаленко М.Ш. Моделирование семантики в базах данных. М.: Наука, 1989. - 287 С.
84. Черняк Л. Intranet - объективная реальность, данная нам// Рынок ценных бумаг, №3, 1997.
85. Чжо З.Е., Чжо З.Л., Наинг Л.А. Разработка методики децентрализованной диагностики и диагностируемости с использованием

модели тестирования// Вести высших учебных заведений Черноземья. 2015. № 3. С. 60-70.

86. Чистяков В.П. Курс теории вероятностей - 5-е изд. М.: Агар, 2000. - 255 С.

87. Шагурина Н. Web-службы: новая парадигма интеграции? / Сетевой журнал №2, М., 2003 – С. 14-17.

88. Шеннон Р. Имитационное моделирование систем. М.: Мир, 1978. - 418 С.

89. Abdalov A.V., Grishakov V.G., Loginov I.V., Spitsyn A.A. Resource planning algorithm for reconfigurable information system development in the case of several sources of resources// AIP Conference Proceedings, 2402, 050012, <https://doi.org/10.1063/5.0071507>.

90. Abdelkrim A., Hock-Koon A., and Oussalah M.C. Chapter 1: Object-Oriented, Component-Based, Agent-Oriented and Service Oriented Paradigms, // Software Architectures 1, Wiley-ISTE, 2014.

91. Afriansyah MF, Somantri M, Riyadi MA (2017) Model of load balancing using reliable algorithm with multi-agent system. IOP Conference Series: Materials Science and Engineering, Volume 190, conference 1.

92. Aghdashi A, Mirtaheri SL (2019) A Survey on Load Balancing in Cloud Systems for Big Data Applications. In: Grandinetti L., Mirtaheri S., Shahbazian R. (eds) High-Performance Computing and Big Data Analysis. TopHPC 2019. Communications in Computer and Information Science, vol 891. Springer, Cham

93. Aglet user manual. <http://aglets.sourceforge.net/>.

94. Aglets. <http://en.wikipedia.org/wiki/Aglets>.

95. Ahmed YE, Adjallah KH, Stock R, Kacem I, Babiker SF. NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring. Comput Indust Eng. 2018;115:17-25.

<http://www.sciencedirect.com/science/article/pii/S0360835217304667>.

96. Ali M, Bagchi S (2019) Design and analysis of distributed load management: Mobile agent. based probabilistic model and fuzzy integrated model. *Appl Intell*, Vol. 49, pp. 3464-3489.

97. Al-Kiyumi RM, Foh CH, Vural S, Chatzimisios P, Tafazolli R. Fuzzy logic-based routing algorithm for lifetime enhancement in heterogeneous wireless sensor networks. *IEEE Trans Green Commun Netw.* 2018;2(2):517-532.

98. Anjomshoa MF, Salleh M, Kermani MP (2015) A Taxonomy and Survey of Distributed Computing Systems. *Journal of Applied Sciences*, Vol. 15, pp. 46-57.

99. Bolnokin V.E., Sorokin S.A., Gorshkov A.V. Algorithms for manufacturing system control under operating conditions change// *International Journal on Information Technologies and Security*, No.3 (vol. 13), 2021, pp. 39-48.

100. Brazier F.M.T., Overeinder B.J., Steen M.V., Wijngaards H.J.E. Generative mobile agent migration in heterogeneous environments// *Proc. of the AIMS Workshop at SAC 2002*, 2002, pp. 101-106.

101. Buragohain C, Agrawal D, Suri S. Power aware routing for sensor databases. In: *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 3 IEEE; 2005:1747-1757.

102. Centola D. The spread of behavior in an online social network// *Science*, 2010, 329(5996):1194-1197.

103. Chakraborty D. et al. Service composition for mobile environments// *Mobile Networks and Applications*, 2005, 10(4), p. 435-451.

104. Chakravorty S., Mendes C.L., Kale L.V. Proactive fault tolerance in MPI applications via task migration// *Lecture Notes in Computer Science*, 2006, Vol. 4297, pp. 485-496.

105. Chierichetti F., Lattanzi S., Panconesi A. Rumor spreading in social networks// Theoret. Comput. Sci., 2011, 412(24):2602-2610.

106. Choi B.K., Rho S., Bettati R. Fast software component migration for applications survivability in distributed real-time systems// Proc. of 7th IEEE Int. Symp. on Object-Oriented Real-Time Distributed Computing, Vienna, Austria, 2004, pp. 269-276.

107. Colizza V., Vespignani A. Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: Theory and simulations// J. Theoret. Biol., 2008, 251(3):450-467.

108. Couturier R, Giersch A, Hakem M (2018) Best effort strategy and virtual load for asynchronous iterative load balancing. Journal of Computational Science. Vol. 26, Pages 118-127.

109. Dasgupta K, Kalpakis K, Namjoshi P. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In: Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, Vol. 3 IEEE; 2003:1948-1953.

110. De Arruda G.F., Rodrigues F.A., Rodriguez P., Cozzo E., Moreno Y. A general Markov chain approach for disease and rumour spreading in complex networks// J. Complex Networks, 2018, 6(2):215-242.

111. Ding M, Cheng X, Xue G. Aggregation tree construction in sensor networks. In: Vehicular Technology Conference, 2003. vtc 2003-fall. 2003 IEEE 58th, Vol. 4 IEEE; 2003:2168-2172.

112. Engelmann C., Vallee G.R., Naughton T., Scott S.L. Proactive fault tolerance Using preemptive migration// Proc. of 17th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing, 2009 pp. 252-257.

113. Fasolo E, Rossi M, Widmer J, Zorzi M. In-network aggregation techniques for wireless sensor networks: a survey. Wireless Commun. 2007;14(2):70-87. <https://doi.org/10.1109/MWC.2007.358967>.

114. Feola G., Butt A. The diffusion of grassroots innovations for

sustainability in Italy and Great Britain: An exploratory spatial data analysis// Geographical J., 2017, 183(1):16-33.

115. Fu S., Xu C. Service migration in distributed virtual machines for adaptive grid computing// Int. Conf. on Parallel Processing, 2005, pp. 358-365.

116. Furno A., Zimeo E. Context-aware Composition of Semantic Web Services// Mobile Networks and Applications, 2014. 19(2), p. 235-248.

117. Gao F., Curry E., Ali M.I., Mileo A. QoS-Aware Complex Event Service Composition and Optimization Using Genetic Algorithms// Lecture Notes in Computer Science, 2014, vol. 8831. DOI: 10.1007/978-3-662-45391-9_28.

118. Gilboa-Freedman G., Hassin R. When Markov chains meet: A continuous-time model of network evolution// Statist. Probability Lett., 2016, 116:131-138.

119. Gorshkov A.V. Data aggregation and network lifetime increase factor: a new algorithm and simulation study// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2023'AS): Proceedings of the XXVII-th International Open Science Conference (Yelm, WA, USA, January 2023). - Yelm, WA, USA: Science Book Publishing House, 2023. – Pp. 123-134.

120. Gorshkov A.V. The problem of network lifetime increase in poorly deterministic networks based on data aggregation trees// Modern informatization problems in simulation and social technologies (MIP-2023'SCT): Proceedings of the XXVIII-th International Open Science Conference (Yelm, WA, USA, January 2023). - Yelm, WA, USA: Science Book Publishing House, 2023. – Pp. 28-29.

121. Gorshkov A.V., Kravets O.Ja., Aksenov I.A., Redkin Yu.V., Atlasov I.V. Architecture of a video analytics system using parallel processing// International Journal on Information Technologies and Security, No 4 (vol. 14), 2022, pp. 3-12.

122. Groenevelt R., Nain P., Koole G. The message delay in mobile ad hoc networks// Performance Evaluation, 2005, 62:210-228.

123. Hajduk M, Sukop M, Haun M (2019) Cognitive Multi-agent Systems : Structures, Strategies and Applications to Mobile Robotics and Robosoccer. In Series of Studies in Systems, Decision and Control.

124. Hakansson A, Hartung R (2013) Agent and Multi-Agent Systems in Distributed Systems - Digital Economy and E-Commerce. Publisher: Springer-Verlag Berlin Heidelberg.

125. Hammerschmidt B.C., Linnemann V. Migratable Web services: increasing performance and privacy in service oriented architectures// International Journal on Computer Science and Information Systems, 2006, 1(1), pp. 42-56.

126. He J, Ji S, Pan Y, Li Y. Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks. IEEE Trans Parallel Distrib Syst. 2014;25(7):1681-1690.

127. Higashino M., Takahashi K., Kawamura T., Sugahara K. Mobile agent migration based on code caching// Proc. of 26th Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA), 2012, pp. 651-656.

128. Hock-Koon A., Oussalah M. Composite service metamodel and auto composition// Journal of Computational Methods in Science and Engineering, 2010, 10, p. 215-229.

129. Hock-Koon A., Oussalah M. Expliciting a composite service by a metamodeling approach// Research Challenges in Information Science (RCIS), 2010 Fourth IEEE Int. Conf. on. 2010.

130. Hsieh H, Chiang M (2019) The Incremental Load Balance Cloud Algorithm by Using Dynamic Data Deployment. J Grid Computing Vol. 17, pp. 553–575.

131. Imon SKA, Khan A, Di Francesco M, Das SK. Energy-efficient

randomized switching for maximizing lifetime in tree-based wireless sensor networks. *IEEE/ACM Trans Netw (TON)*. 2015;23(5):1401-1415.

132. Isella L., Stehlé J., Barrat A., Cattuto C., Pinton J.-F. What's in a crowd? Analysis of face-to-face behavioral networks// *J. Theoret. Biol.*, 2011, 71(1):166-180.

133. Ishikawa F., Yoshioka N., Tahara Y., Honiden S. Mobile agent system for Web services integration in pervasive networks// *Proc. of the 1st Int. Workshop on Ubiquitous Computing*, 2004, pp. 38–47.

134. Jifeng H., Li X., Liu Z., Component-based software engineering// *Theoretical Aspects of Computing–ICTAC*, Springer, 2005, p. 70-95.

135. Jin M, Liu F, Zhou C Rumor spreading: A survey// *2nd Int. Conf. on Artificial Intelligence Engrg. Appl.*, 2017, 263-269.

136. Jothiprakasam S, Muthial C. A lifetime enhancement protocol in cluster-based wireless sensor networks with guaranteed delay. *Int J Commun Syst*. 2019;32(12):e3983. <https://doi.org/10.1002/dac.3983>.

137. Kale PA, Nene MJ. Asymptotic cost estimation for scheduling data aggregation trees in sensor networks. In: *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*; 2019:409-414.

138. Kale PA, Nene MJ. Data aggregation trees with QoS in sensor networks. In: *2019 IEEE 5th International Conference for Convergence in Technology (i2ct)*; 2019:1-5.

139. Kale PA, Nene MJ. Path reestablishment in wireless sensor networks. In: *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*; 2017:1659-1663.

140. Kale PA, Nene MJ. Quality aware data aggregation trees in sensor networks. In: *Intelligent Communication Technologies and Virtual Mobile Networks Springer*; 2019:557-567.

141. Kale PA, Nene MJ. Scheduling of data aggregation trees using local heuristics to enhance network lifetime in sensor networks. *Computer Networks*.

2019;160:51-64.

142. Kuo T-W, Lin KC-J, Tsai M-J. On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NPcompleteness and approximation algorithms. *IEEE Trans Comput.* 2016;65(10):3109-3121.

143. Lee M, Wong VW. An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks. In: *Communications, Computers and Signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on IEEE*; 2005:300-303.

144. Li H. Load Balancing Algorithm for Heterogeneous P2P Systems Based on Mobile Agent. in *IJCA Proceedings on International Conference on Electronics, Information and Communication Engineering (ICEICE) (Foundation of Computer Science, New York, 2011)*, pp. 1446–1449.

145. Liang W, Xu Y, Shi J, Luo J. Aggregate node placement for maximizing network lifetime in sensor networks. *Wirel Commun Mobile Comput.* 2012;12(3):219-235. <https://doi.org/10.1002/wcm.952>.

146. Lin H-C, Li F-J, Wang K-Y. Constructing maximum-lifetime data gathering trees in sensor networks with data aggregation. In: *Communications (ICC), 2010 IEEE International Conference on IEEE*; 2010:1-6.

147. Liu C, Cao G. Distributed monitoring and aggregation in wireless sensor networks. In: *INFOCOM, 2010 Proceedings IEEE*; 2010:1-9.

148. Liu X, Liu A, Wang T, Ota K, Dong M, Liu Y, Cai Z. Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks. *J Parall Distrib Comput.* 2020;135:140-155.

149. Liu X, Zhao M, Liu A, Wong KKL. Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks. *Inf Fusion.* 2020;53:183-195.

150. Liu X. An optimal-distance-based transmission strategy for lifetime maximization of wireless sensor networks. *IEEE Sens J.* 2015;15(6): 3484-3491.

151. Lu Y, Comsa I-S, Kuonen P, Hirsbrunner B. Adaptive data aggregation with probabilistic routing in wireless sensor networks. *Wirel Netw.* 2016;22(8):2485-2499.
152. Madden S, Franklin MJ, Hellerstein JM, Hong W. TAG: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper Syst Rev.* 2002;36(SI):131-146. <https://doi.org/10.1145/844128.844142>.
153. Manzo G., Gabbriellini S., Roux V., M'Mbogori F.N. Complex contagions and the diffusion of innovations: evidence from a small-N study// *J. Archaeological Method Theory*, 2018, 25(4):1109-1154.
154. Mascia M.B., Mills M. When conservation goes viral: The diffusion of innovative biodiversity conservation policies and practices// *Conservation Lett.*, 2018, 11(3):e12442, <https://doi.org/10.1111/conl.12442>.
155. Matsuura H. Maximizing lifetime of multiple data aggregation trees in wireless sensor networks. In: *Network Operations and Management Symposium (NOMS)*, 2016 IEEE/IFIP IEEE; 2016:605-611.
156. Metawei MA, Ghoneim SA, Haggag SM, Nassar SM (2012) Load balancing in distributed multi-agent computing systems, *Ain Shams Engineering Journal*, Vol. 3, pp. 237-249, May 2012.
157. Milling C., Caramanis C., Mannor S., Shakkottai S. Distinguishing infections on different graph topologies// *IEEE Trans. Inform. Theory*, 2015, 61(6):3100-3120.
158. Nehra N, Patel RB, Bhat VK (2008) Load Balancing in Heterogeneous P2P Systems using Mobile Agents. *International Journal of Electrical and Computer Engineering* Vol:2, No:8, pp. 2740-2745.
159. Neogy R., Chowdhury C., Neogy S. Reliability of mobile agents for reliable service discovery protocol in MANET// *International Journal of Wireless & Mobile Networks*, 2011, Vol.3, No. 5, pp. 229-243.
160. Nguyen N-T, Liu B-H, Pham V-T, Luo Y-S. On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data

aggregation trees. Comput Netw. 2016; 105(C): 99-110.
<https://doi.org/10.1016/j.comnet.2016.05.022>.

161. Nowzari C., Preciado V.M., Pappas G.J. Analysis and control of epidemics// IEEE Control Systems Magazine, 2016, 36(1):26-46.

162. Pastor-Satorras R., Castellano C., Van Mieghem P., Vespignani A. Epidemic processes in complex networks// Rev. Modern Phys., 2015, 87(3):925-979.

163. Priya V, Kumar CS, Kannan R (2019) Resource scheduling algorithm with load balancing for cloud service provisioning. Appl Soft Comput Vol. 76, pp. 416–424.

164. Rajagopalan R, Varshney PK. Data aggregation techniques in sensor networks: a survey. IEEE Commun Surv Tut. 2006;8:48-63.

165. Ramos AG, Silva E, Oliveira JF (2018) A new load balance methodology for container loading problem in road transportation. European Journal of Operational Research, Vol. 266(3), pp. 1140–1152.

166. Rogers E.M. Diffusion of Innovations, 4th ed. - Free Press, New York, 1995.

167. Rosa R.E.V., Lucena V.F. Jr. Smart composition of reusable software components in mobile application product lines// Proc. of the 2nd Int. ACM Workshop on Product Line Approaches in Software Engineering. 2011.

168. Sadeghi F, Avokh A. Load-balanced data gathering in internet of things using an energy-aware cuckoo-search algorithm. International Journal of Communication Systems. 2020;33(9):e4385. <https://doi.org/10.1002/dac.4385>.

169. Salehi M. A., H. Deldari, and B. M. Dorri (2009) Balancing load in a computational grid applying adaptive, intelligent colonies of ants, Informatica, vol. 33, no. 2, pp. 159–168.

170. Samolej S. and Rak T. (2009) Simulation and Performance Analysis of Distributed Internet Systems Using TCPNs, Informatica, vol. 33, pp. 405-415.

171. Sattenspiel L., Dietz K. A structured epidemic model incorporating

geographic mobility among regions// Math. Biosciences, 1995, 128(1-2):71-91.

172. Semchedine F, Bouallouche-Medjkoune L, Tamert M, Mahfoud F, Aïssani D (2015) Load balancing mechanism for data-centric routing in wireless sensor networks. J. Comput. Electrical Eng. Vol. 41, pp. 395–406.

173. Shan M, Chen G, Luo D, Zhu X, Wu X. Building maximum lifetime shortest path data aggregation trees in wireless sensor networks. ACM Trans Sensor Netw (TOSN). 2014;11(1):11.

174. Shen X-J et al (2014) Achieving dynamic load balancing through mobile agents in small world P2P networks, Comput. Netw., vol. 75, pp. 134-148, Dec.

175. Simpson G., Clifton J. Testing diffusion of innovations theory with data: Financial incentives, early adopters, and distributed solar energy in Australia// Energy Res. Soc. Sci., 2017, 29:12-22.

176. Spitsyn A.A., Mutin D.I. Architecture of the prototype of the task distribution management system in cloud environments// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2023'AS): Proceedings of the XXIX-th International Open Science Conference (Yelm, WA, USA, January 2023). - Yelm, WA, USA: Science Book Publishing House, 2023.

177. Spitsyn A.A. About optimization approaches to cloud load balancing// Modern informatization problems in simulation and social technologies (MIP-2022'SCT): Proceedings of the XXVII-th International Open Science Conference (Yelm, WA, USA, January 2022). - Yelm, WA, USA: Science Book Publishing House, 2022. – p. 174-183.

178. Spitsyn A.A., Mutin D.I. Approach to creating a hierarchical load balancing strategy in cloud structures// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2021'AS): Proceedings of the XXVI-th International Open Science Conference (Yelm, WA, USA, January 2021). - Yelm, WA, USA: Science Book Publishing

House, 2021. – p. 167-175.

179. Spitsyn A.A., Mutin D.I., Bikkulov I.M., Frantsisko O.Yu., Atlasov I.V. Multi-objective selection of structure variants for a corporate heterogeneous integrated system of information management// International Journal on Information Technologies & Security, № 1 (vol. 13), 2021, p. 27-38. **(WoS)**

180. Sun H., Cheng R., Xiao X., Yan J., Zheng Y., Qian Y. Maximizing social influence for the awareness threshold model// Database Systems for Advanced Applications, Lecture Notes in Computer Science, 2018, vol. 10827 (Springer, Cham), 491-510.

181. Sward R., Boleng J. Service-oriented architecture (SOA) concepts and implementations// ACM SIGAda Ada Letters, 2011, 31(3). DOI: 10.1145/2070336.2070340.

182. Tan HÖ, Körpeolu I. Power efficient data gathering and aggregation in wireless sensor networks. ACM Sigmod Record. 2003;32(4):66-71.

183. Van Steen M, Tanenbaum AS (2016) A brief introduction to distributed systems. Computing. vol. 98, no. 10, pp. 967-1009.

184. Vespignani A. Modelling dynamical processes in complex socio-technical systems// Nature Phys., 2012, 8:32-39.

185. Wang C., Mueller F., Engelmann C., Scott S.L. Proactive process-level live migration in HPC environments// Proc. of the IEEE/ACM Int. Conf. on High Performance Computing, Networking, Storage and Analysis, Austin, TX, USA, 2008, pp. 1-12.

186. Wang J, Gao Y, Liu W, Wu W, Lim S-J. An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. Comput Mater Contin. 2019;58:711-725.

187. Wang J, Gao Y, Wang K, Sangaiah AK, Lim S-J. An affinity propagation-based self-adaptive clustering method for wireless sensor networks. Sensors. 2019;19(11):2579.

188. Wang J, Gao Y, Yin X, Li F, Kim H-J. An enhanced PEGASIS

algorithm with mobile sink support for wireless sensor networks. *Wirel Commun Mobile Comput.* 2018;2018.

189. Wang J, Gao Y, Zhou C, Sherratt RS, Wang L. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns. *Comput Mater Continua.* 2020; 62(2):695-711. <http://www.techscience.com/cmcc/v62n2/38271>.

190. Wang J, Gu X, Liu W, Sangaiah AK, Kim H-J. An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Human-centric Comput Inf Sci.* 2019;9(1):1-14.

191. Wided A. and Okba K. (2019) A novel Agent Based Load Balancing Model for maximizing resource utilization in Grid Computing, *Informatica*, vol. 43, no. 3.

192. Wood T., Ramakrishnan K.K., van der Merwe J. CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines// *Proc. of 7th ACM SIGPLAN/SIGOPS Int. Conf. on Virtual Execution Environments (VEE)*, 2011, pp. 121-132.

193. Wu Y, Mao Z, Fahmy S, Shroff NB. Constructing maximum-lifetime data-gathering forests in sensor networks. *IEEE/ACM Trans Netw.* 2010;18(5):1571-1584.

194. Wu Z. et al. Automatic composition of semantic web services using process and data mediation// *ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume SAIC, Funchal, Madeira, Portugal, 2007*.

195. Xie S, Wang Y. Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wirel Personal Commun.* 2014;78(1):231-246. <https://doi.org/10.1007/s11277-014-1748-5>.

196. Xiong H., Wang P., Bobashev G. Multiple peer effects in the diffusion of innovations on social networks: a simulation study// *J. Innovation Entrepreneurship*, 2018, 7(1):1-18.

197. Xue Y, Cui Y, Nahrstedt K. Maximizing lifetime for data aggregation in wireless sensor networks. *Mobile Netw Appl.* 2005;10(6):853-864.
198. Yildiz HU. Investigation of maximum lifetime and minimum delay trade-off in underwater sensor networks. *Int J Commun Syst.* 2019; 32(7):e3924. <https://doi.org/10.1002/dac.3924>
199. Younes H, Bouattane O, Youssfi M, Illoussamen E (2017) New load balancing framework based on mobile AGENT and ant-colony optimization technique. *Intelligent Systems and Computer Vision (ISCV)*, Fez, 2017, pp. 1-6.
200. Youssfi M, Bouattane O, Bensalah, M (2015) Efficient Load Balancing Algorithm for Distributed Systems Using Mobile Agents , *Advanced Studies in Theoretical Physics* Vol. 9, no. 5, pp.245 – 253.
201. Zgaya H., Hammadi S. Multi-agent information system using mobile agent negotiation based on a flexible transport ontology// *Proc. of the Sixth Int. Joint Conf. on Autonomous Agents and Multi-Agent System*, Hawaii, USA, 2007, pp. 1024-1036.
202. Zhang J. A mobile agent-based tool supporting Web services testing// *Wireless Personal Communications*, 2011, 56, pp. 147-172.
203. Zhang X. et al. Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing// *Mobile Networks and Applications*, 2011. 16(3), p. 270-284.
204. Zhang X., Neglia G., Kurose J., Towsley D. Performance modeling of epidemic routing// *Comput. Networks*, 2007, 51(10):2867-2891
205. Zhou F, Chen Z, Guo S, Li J. Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs. *IEEE Sens J.* 2016; 16(22):8167-8177.
206. Zuo Y., Wang X. Semantic specification of service migration for system survivability// *Proc. of IEEE Int. Conf. on Electro/Information Technology*, Milwaukee, USA, 2014.