

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ  
РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**ИНФОРМАЦИОННЫЕ  
ТЕХНОЛОГИИ**

Межвузовский сборник научных трудов

Рязань 2017

УДК 004

Информационные технологии, межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет, 2017. – 203 с.

ISBN 978-5-7722-0301-9

Публикуются статьи о проблемах использования информационных технологий в науке и образовании.

Сборник рассчитан на научно-педагогических работников вузов и может быть использован студентами и аспирантами инфокоммуникационных и информационных специальностей.

Редакционная коллегия:

Д-р техн. наук, проф. В.П. Корячко (ответственный редактор), д-р техн. наук, проф. Я.Е. Львович (Воронежский государственный технический университет), д-р техн. наук, проф. А.Д. Иванников (Институт проблем проектирования в микроэлектронике Российской академии наук), д-р техн. наук, проф. С.В. Скворцов (РГРТУ), д-р. техн. наук, проф. А.П. Шибанов (РГРТУ), канд. техн. наук А.Н. Сапрыкин (ответственный секретарь).

Рецензенты:

Кафедра ВПМ Рязанского государственного радиотехнического университета (зав. кафедрой, д-р техн. наук, проф. А.Н. Пылькин), д-р техн. наук, проф. Е.А. Саксонов (Московский государственный институт электроники и математики (ТУ)).

**ISBN 978-5-7722-0301-9**

©Рязанский государственный  
радиотехнический университет, 2017

## Содержание

**Апрельский А.А**

Микроконтроллеры AVR на базе RISC архитектуры..... 8

**Вялов А.В.**

Программная реализация алгоритма компоновки электронных средств..... 10

**Гольд М.В.**

Разработка программной модели кольцевой сети ..... 13

**Гостин А.М., Косоруков С.С., Шумилова А.С.**

Алгоритм работы модуля импорта учебного плана образовательного портала вуза..... 17

**Гостин А.М., Псоянц В.Г., Сапрыкин А.Н.**

Единая образовательная платформа адаптированных основных профессиональных образовательных программ..... 21

**Гринченко Н.Н., Клименко Е.А.**

Реализация модели системы передачи цифровых данных с использованием массивно-параллельных вычислений ..... 30

**Гусева А.И.**

Использование системы T-FLEX CAD при разработке модуля управления соединительным устройством..... 35

**Евстафьев Д.А.**

Конструирование системы питания ветроэнергетической установки..... 42

**Жданкина Е.А., Решетова В.С., Софьин Е.А.**

Реализация алгоритмов сжатия изображений на базе вейвлет-преобразования и их сравнение ..... 46

**Журавлев В.А., Орешков В. И.**

Инновационный инжиниринг для объектов инфраструктуры ..... 52

**Киселева В.О.**

Разработка программного модуля, исследующего сетевые характеристики в SDN сетях..... 57

**Косырев А.А.**

Автоматическая корректировка погрешности измерительного устройства ..... 59

**Лунякова М.А., Бакулева М.А.**

Математическая модель результатов первичной диагностики для экспертной системы врача-гематолога ..... 63

**Маркина П.П.**

Программная реализация модели безопасности HRU, содержащая проверку ее состояния безопасности ..... 67

**Мелехин С.А., Юнюшкин Д.С.**

Применение и управление технологией АФАР на беспилотных летательных аппаратах ..... 70

**Найденов А.С.**

Применение сэмплинга в условиях несбалансированности классов ..... 73

**Нгуен А.З., Шибанов А. П.**

Использование алгоритмов разбиения при разрешении конфликтов в системах передачи информации с множественным доступом ..... 76

**Нгуен Н.З., Копейкин Ю.А.**

Построение быстродействующих перемножителей на базе ПЛИС ..... 79

**Нгуен Т.Ф., Шибанов А.П., Фам Х.Л.**

Имитационное моделирование системы передачи измерительной информации ..... 82

**Осин И.Г., Манакин А.А.**

Средства распознавания речи ..... 91

**Перепелкин Д.А., Бышов В.С.**

Адаптивная маршрутизация в программно-конфигурируемых сетях ..... 94

**Перепелкин Д.А., Цыганов И.Ю.**

Комбинированный алгоритм Дейкстры и сегментации в программно-конфигурируемых сетях ..... 97

**Перов А.В.**

Применение параллельного программирования при реализации алгоритмов ..... 101

**Писака П.С.**

Алгоритм привязки файлов телеметрической информации на основе взаимного анализа бортового и стационарного времен ..... 103

**Полилов И.А.**

Комплексные системы безопасности для промышленных помещений ..... 108

**Попов Г.А., Скворцов С.В.**

Анализ возможности параллельного решения задачи коммивояжера ..... 111

**Попов С.В.**

Конструирование блока управления SDN сети и разработка алгоритмов многопутевой маршрутизации ..... 114

**Пюрова Т.А., Скворцов С.В.**

Разработка параллельных алгоритмов поиска данных на платформе CUDA ..... 118

**Ручкин В.Н., Геращенко Е.С., Потапова В.Ю., Тарасов А.С., Токарев А.В.**

Шифрование и дешифрование информации на нейропроцессоре NM6403 ..... 124

**Сапрыкин А.Н., Акинина К.Д.**

Основные характеристики генетического алгоритма компоновки блоков электронно-вычислительных средств ..... 128

**Сапрыкин А.Н., Сапрыкина Е.Н.**

Исследование влияния базовых параметров генетического алгоритма балансировки потоков данных в сетях дата-центров на сходимость ..... 132

**Сапрыкин А.Н., Сапрыкина Е.Н.**

Методы и алгоритмы балансировки потоков данных в сетях дата-центров ..... 139

|   |     |
|---|-----|
| <b>Сапрыкина А.О., Сапрыкин А.Н.</b><br>Основные типы электронного портфолио .....                                  | 142 |
| <b>Сафонов А.Л.</b><br>Взаимодействие сетевых устройств друг с другом в<br>гетерогенных сетях. CWMP .....           | 146 |
| <b>Скуратов О.И., Скуратова П.Э.</b><br>Гибридная ветросолнечная система.....                                       | 150 |
| <b>Соболева А.И., Громов А.Ю.</b><br>Использование OLAP-технологий в системах мониторинга<br>окружающей среды ..... | 153 |
| <b>Соловьев С.А.</b><br>Модуль управления солнечной батареей для слежения<br>за солнцем .....                       | 156 |
| <b>Телялькова М.В., Горин В.С., Соколов В.Г.</b><br>Обзор методов преобразования аналогового сигнала в код.....     | 158 |
| <b>Тишкина Е.В.</b><br>Автоматизированная система для проведения медико-экономической<br>экспертизы.....            | 162 |
| <b>Тришин А.А., Буряков А.П.</b><br>Совершенствование системы обмена данными специального<br>назначения.....        | 165 |
| <b>Тришин А.А., Жарова А.С.</b><br>Особенности моделирования каскадов с трансформаторами на<br>длинных линиях ..... | 169 |
| <b>Фам А.М., Шибанов А.П.</b><br>Разработка программы имитационной модели СМО М/М/1 .....                           | 172 |
| <b>Фам В.Т., Сускин В.В.</b><br>Особенности функционирования фотоэлектрической тепловой<br>панели.....              | 176 |

***Филимонов В.В., Сускин В.В.***

Энергосберегающие осветительные приборы для автомобилей с элементами автоматического управления ..... 180

***Фролов А.С.***

Сравнительный анализ программного обеспечения для работы с единой государственной автоматизированной информационной системой ..... 183

***Холопов Е.Ю.***

Подавление случайных геометрических деформаций в реальном времени на базе ПЛИС..... 187

***Челяпова Д.В.***

3D-моделирование камеры соляного тумана..... 190

***Чикалкин В.С.***

Адаптивный алгоритм работы системы синхронного множественного доступа ..... 195

***Шипилов А.А.***

Особенности изготовления печатных плат с использованием 3D принтера ..... 197

**А.А. АПРЕЛЬСКИЙ**

Рязанский государственный радиотехнический университет

**МИКРОКОНТРОЛЛЕРЫ AVR НА БАЗЕ RISC АРХИТЕКТУРЫ**

*Рассматривается AVR микроконтроллер, с возможностью считывания файлов формата MP3 с USB носителя и передачи аналогового сигнала на внешние динамики.*

Однокристалльные микроконтроллеры находят широкие применение в самых разнообразных сферах: от измерительных приборов, фотоаппаратов и видеокамер, принтеров, сканеров и копируемых аппаратов до изделий электронных развлечений и всевозможной домашней техники.

Со времени появления первых микропроцессоров в 1970-х годах их сложность постоянно возрастала за счет появления новых аппаратных решений и добавления новых команд, предназначенных для решения новых задач. Так постепенно сложилась архитектура, получившая впоследствии название CISC (Complete Instruction Set Computers – компьютеры со сложным набором команд). В дальнейшем обозначилось и нашло активное развитие еще одно направление: архитектура RISC (Reduced Instruction Set Computers – компьютеры с сокращенным набором команд). Именно к этой архитектуре относятся микроконтроллеры AVR.

Удачное сочетание RISC-архитектуры «ядра», обеспечивающей высокую производительность, с широким набором команд, Flash-памятью для программ быстро выдвинуло микроконтроллеры AVR на передовые позиции.

На смену микроконтроллерам первых семейств (Tiny и Classic) пришло новое поколение микроконтроллеров (Mega). Сохранив программную преемственность, микроконтроллеры Mega приобрели новые свойства: пониженное напряжение питания (до 2,7 В) и энергопотребление, повышенное быстродействие (до 16 МГц) и объем Flash-памяти (до 128 Кбайт). Вслед за 8-разрядными микроконтроллерами появились 32-разрядные микроконтроллеры AVR32.

Одновременно были созданы программные продукты и технические средства, поддерживающие разработку программ для микроконтроллеров. Это, прежде всего, фирменный пакет фирмы Atmel AVR Studio, свободно распространяемый в сети Internet.

В работе рассмотрены основные этапы разработки и программирования устройства для воспроизведения медиа файлов (в данном случае музыкальных файлов формата MP3) с внешнего носителя USB. Это



устройство основано на микроконтроллере ATmega128 и декодере цифрового сигнала в аналоговый VS1011E.

Вышеупомянутое устройство для считывания и воспроизведения медиа файлов, будет монтироваться в автомобильный магнитофон, не имеющий возможности воспроизведения файлов формата MP3 с внешнего носителя.

Первоначальным этапом разработки, будет являться - выбор микроконтроллера, который будет осуществлять нужную функцию. В данном случае требуемую операцию будет выполнять микроконтроллер ATmega128, так как он является наиболее распространенным в решении подобных задач.

На втором этапе разработки, в программе Sprint Layout будет нарисована схема платы.

Третий этап, включает в себя разработку платы и перенос нарисованных дорожек субтрактивным методом.

Четвертый этап - монтирование элементов на печатную плату.

Пятый, заключительный этап - программирование готового устройства посредством языка ассемблера в ППП AVR Studio.

Принцип работы данного устройства, будет заключаться в следующем:

- на первом этапе нужно осуществить форматирование USB носителя объемом до 16 Gb под файловой системой FAT32.

- на втором этапе на подготовленный USB носитель записываются MP3 файлы.

- на третьем этапе USB носитель с MP3 файлами подключается к уже запрограммированному, путем среды разработки AVR Studio, готовому устройству.

- на четвертом этапе микроконтроллер ATmega128 обрабатывает полученную с USB носителя цифровую информацию и передаёт её в виде цифровых сигналов на декодер VS1011E, который, в свою очередь, выходной аналоговый сигнал передаёт на усилитель магнитофона и далее на внешние динамики.

Готовое устройство будет полезно для использования в автомобилях, имеющих в своей встроенной аппаратуре магнитофон, который не имеет возможности воспроизведения музыкальных файлов с внешнего носителя USB.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. В.Я.Хартов. Микроконтроллеры AVR: Москва, Издательство МГТУ им. Н.Э. Баумана, 2007.

**А.В. ВЯЛОВ**

Рязанский государственный радиотехнический университет

## **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА КОМПОНОВКИ ЭЛЕКТРОННЫХ СРЕДСТВ**

*Рассматриваются подход и принципы объектно-ориентированного программирования, применённые в разработке программной реализации ряда существующих алгоритмов компоновки электронных средств в перспективе их усовершенствования.*

Ввиду того, что в настоящее время развитие электронной промышленности и рост производительности вычислительной техники идёт экстенсивным путём, всё так же имеет высокую важность вопрос большей оптимизации алгоритмов, применяемых на всех этапах разработки ЭС. Одной из разновидностей являются алгоритмы компоновки элементов ЭС, отвечающие за оптимальное распределение по отдельным блокам всего многообразия модулей устройства в ограниченном пространстве. С учётом того, что устройства становятся всё сложнее, важнее становится вопрос оптимального использования доступного пространства, и, следовательно, актуален вопрос разработки способов его осуществления.

Существующие алгоритмы компоновки можно разделить на следующие категории:

- 1) Точные алгоритмы;
- 2) Эвристические;
- 3) Приближённые.

Также алгоритмы компоновки конструктивных узлов классифицируются по критериям оптимизации, по структуре вычислительной процедуры или по ограничениям на формирование узлов.

Из критериев оптимизации основными являются:

- 1) Минимум числа межузловых соединений;
- 2) Минимальная длина межузловых соединений.

В свою очередь, ограничениями являются:

- 1) Количество элементов в узле;
- 2) Число внешних выводов на узле.

Относительно процедуры вычисления алгоритмы компоновки конструктивных узлов можно разделить на:

- 1) Итерационные;
- 2) Последовательные;
- 3) Параллельно-последовательные.

В последнее время наибольшее распространение получили приближенные алгоритмы компоновки (итерационные, последовательные, смешанные).

При использовании последовательных алгоритмов сначала по определенному правилу выбирают вершину графа, затем осуществляют последовательный выбор вершин (из числа нераспределенных) и присоединение их к формируемой части графа. После образования первой части переходят ко второму и так далее до получения желаемого разделения исходного графа.

Была поставлена цель разработать программу, которая предоставляла бы возможность реализовать работу того или иного запрограммированного алгоритма, а также давала возможность сравнить результаты и производительность того или иного алгоритма по отношению к другим. Таким образом, разработка подобной программы позволяет не только проанализировать эффективность алгоритма, но и на стадии разработки усовершенствованного алгоритма проверять его эффективность посредством тестовых проверок и наглядного анализа результатов.

Для имплементации программы было решено использовать следующие технологии и программные средства:

1) Язык программирования – Java 8. Одной из главных особенностей является поддержка лямбда-выражений для работы со структурами данных в функциональном стиле;

2) Фреймворк для внутреннего тестирования функционала программы – TestNG. Так как в процессе разработки производятся изменения в функционале тех или иных модулей, необходимо периодически проверять работу каждого модуля. За это отвечают модульные тесты, написанные с использованием данного фреймворка.

Проект имеет следующую структуру: в программе реализован класс-генератор данных, который производит случайное множество связанных модулей с заданным ограничением количества связей и самих модулей. Работа с множествами осуществляется посредством использования классов-сущностей: Module и Block - каждый блок содержит в себе список модулей, который будет заполнен в результате компоновки. Также присутствуют следующие классы Helper, отвечающий за общий для всех алгоритмов функционал, и Timer, предоставляющий возможность регистрировать время выполнения алгоритма.

Работа приложения была проверена на запрограммированном алгоритме Гэмблина. Данный алгоритм достаточно прост и представляет собой последовательный алгоритм компоновки с учётом ограни-

чений по количеству внешних связей блока и количеству модулей внутри блока.

При запуске программы генерируется произвольное множество объектов по заданным параметрам:

- 1) Максимальное количество возможных связей на модуль;
- 2) Количество модулей в множестве.

Также задаются ограничения:

- 1) Максимум модулей на блок;
- 2) Максимум внешних связей на блок.

В результате генерируется множество модулей, в консоли выводится подробная информация по массиву. После этого производится работа алгоритма, которая для 50 модулей и ограничения в 10 максимально допустимых внешних связей на блок занимает, в среднем, 0.2 секунды и создаёт около 10 блоков. Ввиду последовательной работы алгоритма это ожидаемый результат. Существуют пути усовершенствования алгоритма:

1) Итерационное последовательное преобразование блоков (перестановка модулей) с учётом связности модулей с блоком в случае, если после первой итерации было достигнуто ограничение по количеству связей;

2) Взятие первого модуля в блок на основе вычисления оценки его степени связности – количество внешних связей в рамках его  $N$  родственных модулей, где  $N$  – обозначенное ограничение по количеству модулей в блоке.

Результаты эффективности внесённых в алгоритм изменений впоследствии можно регистрировать посредством измерения статистики. Программа предполагает программирование каждого алгоритма как отдельного класса, представляющего собой чёрный ящик, получающий множество элементов и возвращающий множество блоков с распределёнными по ним исходными элементами. Данный подход позволяет использовать копии алгоритма, внедрять в них модификации и отслеживать статистику по всем модификациям.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Горин В.С. Повышение эффективности последовательных алгоритмов компоновки // Вестник РГРТУ. № 54. – Рязань, 2015. С. 104-108.
2. Селютин В.А. Машинное конструирование электронных устройств. – М.: Сов. Радио, 1977.

М.В. ГОЛЬД

Рязанский государственный радиотехнический университет

## РАЗРАБОТКА ПРОГРАММНОЙ МОДЕЛИ КОЛЬЦЕВОЙ СЕТИ

*Рассматривается задача разработки программной модели сети Token Ring с использованием средств объектно-ориентированного языка программирования C# и программной платформы .NET Framework.*

Сеть Token Ring была изначально разработана компанией IBM в 1984 году. Стандарт IEEE 802.5 появился позже сети Token Ring и имеет незначительные отличия. В частности, стандарт не оговаривает конкретную топологию сети и тип канала передачи данных, в то время как в спецификации Token Ring явно указана топология «звезда» и витая пара в качестве линий связи. Общая схема сети представлена на рис. 1.

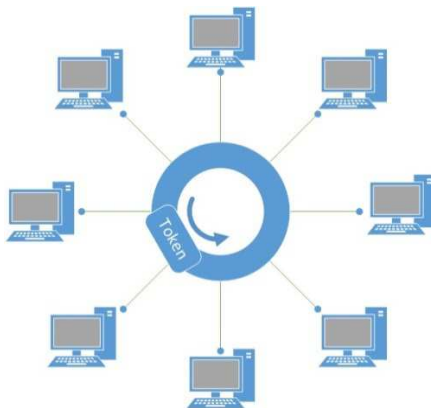


Рис. 1 – Схема сети Token Ring.

Метод доступа Token Ring основан на передаче от узла к узлу специального кадра – токена, или маркера доступа, при этом только узел, владеющий токеном, может передавать свои кадры в кольцо. Для рабочих станций установлено максимальное время удержания токена, по истечении которого станция обязана передать маркер дальше по сети [1]. Данный подход исключает такие ситуации, как неопределенное время ожидания доступа к среде передачи данных. Если время удержания умножить на количество станций, то полученное значение будет максимальным временем передачи маркера по сети.

Структура маркера представлена на рис. 2.

**Маркер**

|                      |    |    |    |
|----------------------|----|----|----|
| <i>Поле</i>          | SD | AC | ED |
| <i>Размер (байт)</i> | 1  | 1  | 1  |

**Кадр данных**

|                      |    |    |    |    |    |    |      |     |    |    |
|----------------------|----|----|----|----|----|----|------|-----|----|----|
| <i>Поле</i>          | SD | AC | FC | DA | SA | RI | Info | FCS | ED | FS |
| <i>Размер (байт)</i> | 1  | 1  | 1  | 6  | 6  | 0  | 0    | 4   | 1  | 1  |

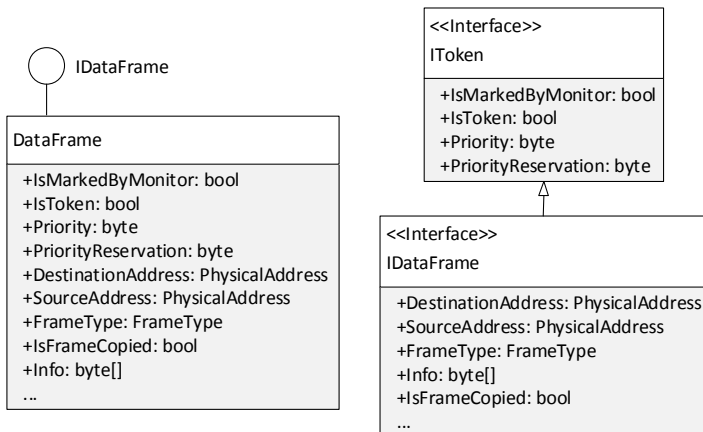
**Рис. 2 – Структура маркера и кадра данных.**

При «пустом» состоянии маркер состоит из начального (Starting Delimiter – SD) и конечного (Ending Delimiter – ED) ограничителей, которые обозначают границы маркера, и поля управления доступом (Access Control – AC), которое содержит значение приоритета, бит свободного маркера и бит прохождения через монитор.

Если станция загружает в пустой маркер свои данные, то добавляются дополнительные поля, которые содержат тип кадра (пользовательский или служебный), адрес получателя, адрес источника, передаваемые данные и другую необходимую информацию.

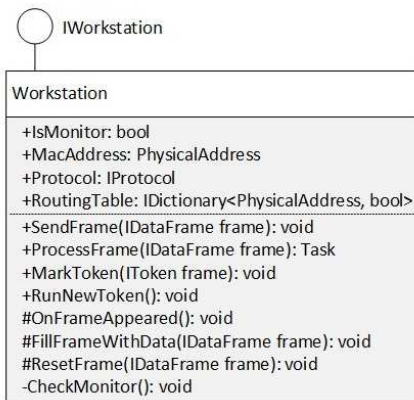
На объектно-ориентированном языке программирования задача представления маркера решается с помощью выделения интерфейсов (Рис. 3). Это позволяет добавлять высокоуровневые операции (обработка, отправка, прием, создание, удаление и др.), оперируя интерфейсами и не привязываясь к конкретным классам. Подобный подход так же применяется и для других объектов (рабочая станция, концентратор, канал передачи данных и др.), что обеспечивает хорошую гибкость и сопровождаемость программы.

Свойства класса DataFrame соответствуют полям структуры кадра данных. Для удобства работы с классом эти поля представлены не в виде совокупности байтов (что характерно для реального маркера), а разделены на свойства с простейшими типами данных (например, биты признаков соответствуют свойствам с булевым типом) и некоторыми сложными типами, такими как PhysicalAddress и FrameType.

Рис. 3 – UML-представление класса **DataFrame**.

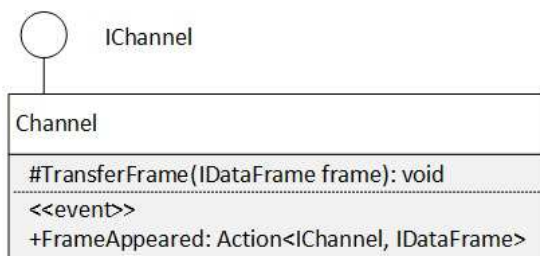
Модель рабочей станции (Рис. 4) содержит методы для отправки (**SendFrame**) и обработки (**ProcessFrame**) кадров, заполнения маркера данными (**FillFrameWithData**), очистки кадра (**ResetFrame**), обработчик события появления кадра в канале (**OnFrameAppeared**), а также признак монитора (**IsMonitor**) и характерные для него методы: проставление метки в маркер (**MarkToken**) и создание нового маркера (**RunNewToken**) на случай, если в сети произошел сбой.

В поле **Protocol** хранится информация о максимальном времени удержания кадра и о том, используется ли механизм приоритетов. Поле **RoutingTable** содержит мас-адреса рабочих станций сети.

Рис. 4 – UML-представление класса **Workstation**.

Если модель создается на основе спецификации IBM Token Ring, то обязательно наличие концентратора. Логика его работы заключается в прослушивании подключенных к нему линий связи и передачи кадра последовательно от одной станции к другой по очереди. Возможна реализация более сложной логики, когда перед отправкой кадра сперва анализируется состояние рабочей станции и если станция вышла из строя, то кадр передается следующей.

Модель канала передачи данных (Рис. 5) достаточно просто реализуется с помощью метода TransferFrame, который инициирует событие FrameAppeared. Узлы сети, подключенные к каналу, подписывают свои обработчики на это событие и выполняют необходимые действия при возникновении кадров.



**Рис. 5 – UML-представление класса Channel.**

Это основные сущности, предложенные для моделирования процесса работы кольцевой сети. Они могут быть изменены как в сторону усложнения, так и в сторону упрощения в зависимости от назначения программы моделирования. Например, если целью моделирования является наглядное представление процесса работы сети, то достаточно использовать сущности типа Кадр данных и Рабочая станция, а также дополнительный класс, предназначенный для управления процессом моделирования (запуск и остановка, масштаб времени и т.д.). Если необходима гибкость и расширяемость программы, то стоит посмотреть в сторону выделения интерфейсов. Если помимо визуальной демонстрации функционирования сети необходим анализ характеристик канала передачи данных, то следует подумать о добавлении классов для сбора статистической информации.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. – СПб.: Питер, 2010. – с. 944.



**А.М. ГОСТИН, С.С. КОСОРУКОВ, А.С. ШУМИЛОВА**  
Рязанский государственный радиотехнический университет

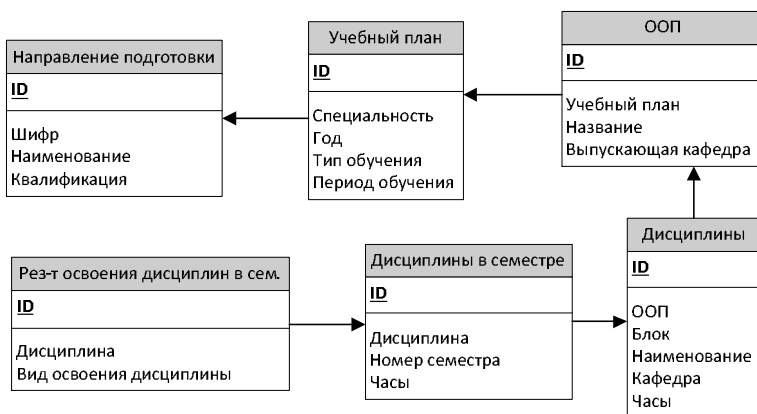
## АЛГОРИТМ РАБОТЫ МОДУЛЯ ИМПОРТА УЧЕБНОГО ПЛАНА ОБРАЗОВАТЕЛЬНОГО ПОРТАЛА ВУЗА

*В статье описан алгоритм импорта учебного плана в формате CSV в базу данных ЭИОС ВУЗа.*

В соответствии с федеральными образовательными стандартами, каждый обучающийся в течение всего периода обучения должен быть обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде (ЭИОС) университета. Центральную часть ЭИОС должна занимать информационная образовательная среда, реализованная на базе образовательного портала. Для функционирования образовательного портала необходимо внести информацию об образовательных программах, реализуемых в ВУЗе. Эти данные доступны в рабочих учебных планах, формируемых соответствующим подразделением ВУЗа (учебное методическое управление) в формате электронных таблиц. [1]

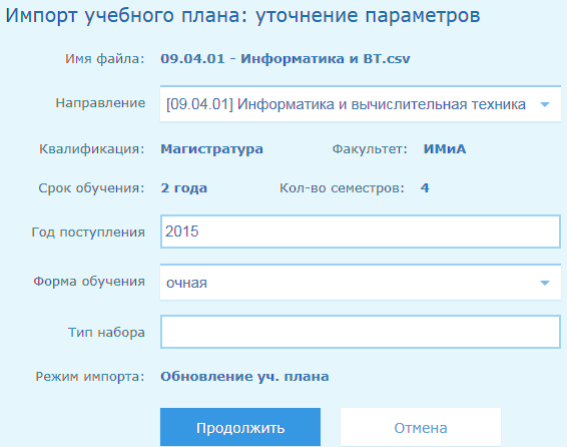
Алгоритм работы модуля импорта можно представить следующим образом:

1. После загрузки файла формата CSV производится его парсинг модулем импорта и формируется массив данных, логическая структура которого представлена на рис. 1. Определяется тип загрузки - первоначальный или обновление.



**Рис. 1 – Логическая структура данных.**

Пользователю отображается окно уточнения параметров, в котором, при необходимости, доступно изменение основных параметров учебного плана (Рис. 2).



Импорт учебного плана: уточнение параметров

Имя файла: 09.04.01 - Информатика и ВТ.csv

Направление: [09.04.01] Информатика и вычислительная техника

Квалификация: Магистратура      Факультет: ИМИА

Срок обучения: 2 года      Кол-во семестров: 4

Год поступления: 2015

Форма обучения: очная

Тип набора:

Режим импорта: Обновление уч. плана

Продолжить      Отмена

**Рис. 2 – Интерфейс уточнения параметров импортированного учебного плана.**

2. На следующем шаге пользователю отображается перечень ООП и дисциплин, подлежащих освоению в данной образовательной программе. Перечень дисциплин содержит данные о количестве часов, выделяемых на различные виды занятий, номера семестров, виды промежуточного контроля по дисциплинам и т.д. Если производится обновление данных об учебном плане, будут отображены текущие хранящиеся в БД портала данные об ООП и дисциплинах с цветовой индикацией обновляемых данных (Рис.3). Для контроля импортируемого учебного плана в интерфейсе отображаются информационные сообщения с примечаниями и ошибками.

Импорт учебного плана: проверка данных

Имя файла: 02.03.03\_D4\_Nabor2016 (21-04-2017 13-51-27).xlsx

Режим импорта: Обновление уч. плана

Направление: [02.03.03] Математическое обеспечение и администрирование информационных систем

Квалификация: Бакалавриат Факультет: ФВТ

Срок обучения: 4 года Кол-во семестров: 8

Год поступления: 2016 Форма обучения: очная

| Дисциплины        |               | Часов в неделю (по семестрам) |             |   |     |     |   |   |   |   |     |
|-------------------|---------------|-------------------------------|-------------|---|-----|-----|---|---|---|---|-----|
| Для всех профилей | Тип           | Зач. ед.                      | Всего часов | 1 | 2   | 3   | 4 | 5 | 6 |   |     |
| Иностранный язык  | Базовая часть | * 8                           | 288         | 2 | 3   | 2   | 3 | 2 | 3 | 2 | Экз |
| Иностранный язык  | Базовая часть | 8                             | 288         | 2 | 3   | 2   | 3 | 2 | 3 | 2 | Экз |
| История           | Базовая часть | * 4                           | 144         |   | 3   | Экз |   |   |   |   |     |
| История           | Базовая часть | 4                             | 144         |   | 3   | Экз |   |   |   |   |     |
| Философия         | Базовая часть | * 4                           | 144         | 3 | Экз |     |   |   |   |   |     |
| Философия         | Базовая часть | 4                             | 144         | 3 | Экз |     |   |   |   |   |     |

Рис. 3 – Интерфейс импорта учебного плана.

При дублировании дисциплин в учебном плане (совпадают название и тип) ячейки подсвечиваются красным. Также подсвечиваются красным строки, отсутствующие в импортируемом учебном плане — они будут удалены. Если к дисциплине привязаны оценки, строка будет помечена значком предупреждения (красный треугольник). В этом случае ее удаление будет невозможно (Рис. 4).

|   |                   |          |             |   |   |   |   |   |   |   |     |
|---|-------------------|----------|-------------|---|---|---|---|---|---|---|-----|
| Устройства ГФС  | Вариативная часть | 5        | 180         |   |   |   |   |   |   |   |     |
| Устройства ГФС  | Вариативная часть | 5        | 180         |   |   |   |   |   |   |   |     |
| Устройства ГФС  | Вариативная часть | 5        | 180         |   |   |   |   |   |   |   |     |
| Цифровая обработка сигналов   | Вариативная часть | 3        | 108         |   |   |   |   |   |   |   |     |
| Цифровая обработка сигналов   | Вариативная часть | 3        | 108         |   |   |   |   |   |   |   |     |
| △ Иностранный язык  | Базовая часть     | 8        | 288         | 2 | 3 | 2 | 3 | 2 | 3 | 2 | Экз |
| Электроника   | Базовая часть     | 4        | 144         |   |   |   |   |   | 4 |   | Экз |
| Устройства ПОС  | Вариативная часть | 5        | 180         |   |   |   |   |   |   |   |     |
| Профиль "Радиотехнические системы локации, навигации и телевидения" | Тип               | Зач. ед. | Всего часов | 1 | 2 | 3 | 4 |   |   |   |     |

Рис. 4 – Дублируемые и удаляемые дисциплины.

Зеленым помечаются ячейки, значения которых будут обновлены. Зеленым помечается вся строка, если дисциплина будет добавлена. Это означает, что ее либо нет в существующем учебном плане, либо название и тип не совпадают с ее значениями в базе. В этом случае нужно обратить внимание на красные строки удаляемых дисциплин — возможно обнаружена опечатка в названии дисциплины или другая ошибка.

Желтым помечаются ячейки для дисциплин по выбору, значения которых были скопированы из строки дисциплины соответствующей

вариативной группы. Красным помечаются ячейки при наличии ошибок в типе аттестации: хотя бы в одном семестре должен быть тип аттестации; если в семестре есть экзамен или зачет, то число часов не должно быть пустым (Рис. 5).

|   |                               |   |     |   |     |   |   |   |   |
|---|-------------------------------|---|-----|---|-----|---|---|---|---|
| Основы программирования на С++                        | Дисциплины по выбору студента | 2 | 72  |   |     |   |   | 2 | 3 |
| Программирование радиотехнических задач               | Дисциплины по выбору студента | 2 | 72  |   | Экз |   |   | 2 | 3 |
| Программирование радиотехнических задач               | Дисциплины по выбору студента | 2 | 72  |   |     |   |   | 2 | 3 |
| Основы теории колебаний в радиотехнике                | Дисциплины по выбору студента | 3 | 108 | 3 | 3   | 3 | 3 | 3 | 3 |
| Основы теории колебаний в радиотехнике                | Дисциплины по выбору студента | 3 | 108 |   |     |   |   | 3 | 3 |
| Методы нелинейных колебаний в информационных системах | Дисциплины по выбору студента | 3 | 108 |   |     |   |   | 3 | 3 |
| Методы нелинейных колебаний в информационных системах | Дисциплины по выбору студента | 3 | 108 |   |     |   |   | 3 | 3 |

**Рис. 5 – Добавление и обновление дисциплин.**

3. По результатам импорта будет отображено информационное сообщение.

Разработанный модуль позволяет работать с рабочими учебными планами разных уровней образования и направлений подготовки, исключать ошибки и несоответствия, что позволяет автоматизировать процесс ввода данных в ЭИОС.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Разработка парсера рабочих учебных планов с применением принципов функционально-реактивного программирования на языке PHP / [Гостин А.М., Гостин П.А., Шумилова А.С.] // Информационные технологии, межвузовский сборник научных трудов. – Рязань: РГРТУ, 2016. – С. 32-35 с.

**А.М. ГОСТИН, В.Г. ПСОЯНЦ, А.Н. САПРЫКИН**  
Рязанский государственный радиотехнический университет

## **ЕДИНАЯ ОБРАЗОВАТЕЛЬНАЯ ПЛАТФОРМА АДАптиРОВАННЫХ ОСНОВНЫХ ПРОФЕССИОНАЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ**

*Рассматривается разработанный прототип образовательной платформы аналитического сопровождения сети ресурсных учебно-методических центров по обучению инвалидов на базе образовательных организаций высшего образования.*

В центре новых информационных технологий РГРТУ в рамках научной работы по заданию Министерства образования и науки РФ был разработан прототип единой образовательной платформы адаптированных основных профессиональных образовательных программ (ОПОП) по нозологиям и укрупненным группам специальностей и направлений (далее — образовательная платформа), предназначенный для экспертно-аналитического сопровождения сети ресурсных учебно-методических центров по обучению инвалидов на базе образовательных организаций высшего образования.

Предполагается, что образовательная платформа будет обеспечивать работу следующих категорий пользователей:

- администратор вуза;
- оператор проекта;
- администратор Министерства;
- незарегистрированный пользователь.

### **Состав прототипа образовательной платформы**

Прототип образовательной платформы построен на базе следующих модулей, логически и функционально связанных друг с другом:

- модуль ввода индивидуальных данных;
- модуль построения и экспорта сводных отчетов;
- личные кабинеты вузов, оператора проекта и Министерства;
- интерфейс образовательной платформы для незарегистрированных пользователей.

Взаимодействие пользователей с веб-приложением прототипа образовательной платформы осуществляется с помощью браузера Mozilla Firefox, Google Chrome или аналогичного.

### **Функции образовательной платформы**

Прототип образовательной платформы реализует следующий набор функций:

- Добавление, удаление, редактирование записей реестра адаптированных ОПОП. Валидация заполнения полей.
- Настройка прав доступа категорий пользователей: вуз, оператор проекта (Исполнитель), Министерство (Заказчик).
- Формирование отчетов по адаптированным ОПОП исходя из критериев:
  - наименования, статуса и географического расположения вуза;
  - направления подготовки (УГСН);
  - наименования и уровня ОПОП;
  - формы и сроков обучения;
  - нозологической категории;
  - статуса адаптированной ОПОП.
- Авторизация категорий пользователей в личном кабинете.
- Загрузка и категоризация ОПОП по направлениям подготовки, нозологиям, вузам и статусу.
- Модерирование и изменение статуса размещенных ОПОП.
- Загрузка методических указаний по освоению ОПОП.
- Публикация мультимедийного контента образовательной платформы, описывающего условия обучения.
- Экспорт отчетов, формируемых образовательной платформой.

#### **Модуль ввода индивидуальных данных**

Модуль предназначен для ввода, редактирования и удаления следующих данных:

- данные о вузах, размещающих ОПОП, необходимые для каталогизации размещаемых ОПОП: краткое и полное наименование вузов, географическое расположение вуза, общие сведения о вузе, данные о кадровом персонале, курирующем обучение лиц с ограниченными возможностями здоровья (ОВЗ), условия для включения лиц с ОВЗ в образовательный процесс;
  - регистрационные данные категорий пользователей, имеющих авторизованный доступ к функциям системы;
  - адаптированные основные образовательные профессиональные программы, каталогизированные по УГСН, уровню подготовки, форме обучения, сроку обучения, нозологической категории.

#### **Модуль построения и экспорта сводных отчетов**

Модуль обеспечивает формирование отчетов по адаптированным

ОПОП исходя из критериев:

- наименования, статуса и географического расположения вуза;
- УГСН;
- наименования и уровня адаптированной ОПОП;
- формы и срока обучения;
- нозологической категории;
- статуса адаптированной ОПОП.

Отчеты могут экспортироваться в формат, совместимый с Excel, из личных кабинетов оператора проекта и Министерства.

#### **Личный кабинет вуза**

Личный кабинет вуза обеспечивает доступ назначенного администратора вуза к списку направлений подготовки, адаптированных ОПОП, формированию записей реестра ОПОП, просмотру методических рекомендаций, публикации мультимедийного контента, описывающего условия обучения, формированию отчетности (Рис. 1).

РЕЕСТР  
АККРЕДИТОВАННЫХ  
ОБРАЗОВАТЕЛЬНЫХ  
ПРОГРАММ
Образовательные программы    О проекте    Личный кабинет    Выход

### Редактирование данных ВУЗа

|                          |  |
|--------------------------|--|
| Полное наименование      | ФГБОУ ВО «Московский государственный психолого-педагогический университет» |
| Сокращенное наименование | ФГБОУ ВО.МГППУ   |
| Псевдоним url адреса     | mgpu   |
| Округ                    | ▼  |
| Регион                   | Республика Бурятия ▼   |
| Официальный сайт         | http://rgsu.net/   |
| Логотип                  | <input type="button" value="Обзор..."/> файл не выбран.                    |

#### Контактное лицо для связи

|                               |  |
|-------------------------------|--|
| Фамилия Имя Отчество          | Алехина Светлана Владимировна  |
| Должность                     | Проректор по инклюзивному образованию  |
| Фотография                    | <input type="button" value="Обзор..."/> файл не выбран.                                    |
| Ученая степень, ученое звание | Кандидат психологических наук  |
| Общественный статус           | Почетный работник общего образования РФ, эксперт ЮНИСЕФ в области инклюзивного образования |
| Телефон                       | 89345467878  |
| E-mail                        | btv@t.ru   |

#### Общие сведения

В МГППУ осуществляется подготовка кадров высшей квалификации с применением дистанционных образовательных технологий. В основу такой подготовки заложен большой опыт создания виртуальных лабораторий, тренажеров и симуляторов, съемки учебных

**Рис. 1 – Интерфейс личного кабинета вуза – редактирование сведений о вузе.**

Функции, доступные назначенному администратору вуза из личного кабинета вуза:

- просмотр и редактирование сведений о вузе, отображаемых по ссылке на наименовании вуза, реализующего образовательную программу в списке образовательных программ: краткое и полное наименование вуза; географическое расположение вуза; общие сведения о вузе; данные о кадровом персонале, курирующем обучение лиц с ОВЗ, условия для включения лиц с ОВЗ в образовательный процесс, имеющиеся в вузе;
- просмотр и редактирование сведений об ОПОП, реализуемых данным вузом: название программы; УГСН; направление; нозологиче-



ская группа, для которой адаптирована данная программа; уровень образования; форма обучения; выпускающая кафедра; заведующий выпускающей кафедры; руководитель программы; описание программы в текстовом виде, включающее возможность добавления мультимедийных элементов в описание; требования к абитуриентам; загрузка файла, содержащего описание программы;

- изменение статуса размещенных в реестре ОПОП, реализуемых данным вузом (опубликовано, черновик, отправить на рассмотрение, отклонено);

- просмотр методических рекомендаций.

#### **Личный кабинет оператора проекта**

Личный кабинет оператора проекта обеспечивает доступ назначенного оператора образовательной платформы к списку вузов, к функциям назначения администратора вуза, редактирования направлений подготовки, модерированию записей реестра адаптированных ОПОП, формирования отчетов.

Функции, доступные назначенному оператору проекта из личного кабинета оператора проекта:

- просмотр, добавление, удаление и редактирование данных о вузах: полное наименование вуза; географическое расположение вуза (округ, регион); контактные данные администратора вуза;

- модерирование записей реестра адаптированных ОПОП: просмотр записей ОПОП, представленных для рассмотрения администраторами вузов; изменение статуса размещенных ОПОП (отклонено, опубликовано);

- просмотр, добавление и удаление записей методических рекомендаций: название методических рекомендаций; текстовое описание методических рекомендаций; файлы с методическими рекомендациями;

- формирование, просмотр и экспорт отчетов в формат, совместимый с Excel: распределение ОПОП, адаптированных по нозологиям, по вузам и регионам; распределение ОПОП по УГСН;

- формирование и просмотр отчетов в виде инфографики: распределение адаптированных основных профессиональных образовательных программ по федеральным округам и регионам (карта); распределение адаптированных основных профессиональных образовательных программ по регионам (круговая диаграмма); распределение адаптированных основных профессиональных образовательных программ по регионам (столбчатая диаграмма), таблица распределения адаптированных образовательных программ по нозологиям.

Формирование отчетов по адаптированным ОПОП производится

исходя из критериев:

- наименование, статуса и географическое расположение вуза;
- направление подготовки (УГСН);
- наименование и уровня ОПОП;
- форма и срок обучения;
- нозологическая категория;
- статус адаптированной ОПОП.

### Личный кабинет Министерства

Личный кабинет Министерства обеспечивает доступ администратора Министерства к отчетности, формируемой образовательной платформой (Рис. 2).

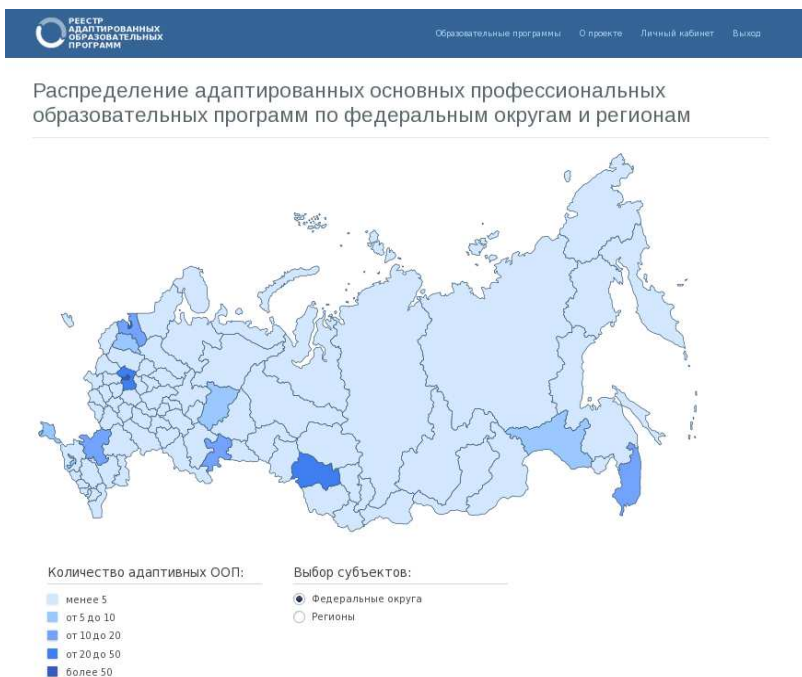


Рис. 2 – Интерфейс личного кабинета Министерства.

Отчеты могут экспортироваться в формат, совместимый с Excel. Функции, доступные назначенному администратору Министерства из личного кабинета оператора проекта:

- формирование, просмотр и экспорт отчетов в формат, совме-

стимый с Excel: распределение ОПОП, адаптированных по нозологиям, по вузам и регионам; распределение ОПОП по УГСН;

- формирование и просмотр отчетов в виде инфографики: распределение адаптированных основных профессиональных образовательных программ по федеральным округам и регионам (карта); распределение адаптированных основных профессиональных образовательных программ по регионам (круговая диаграмма); распределение адаптированных основных профессиональных образовательных программ по регионам (столбчатая диаграмма), таблица распределения адаптированных образовательных программ по нозологиям.

### **Интерфейс образовательной платформы для незарегистрированных пользователей**

Интерфейс образовательной платформы для незарегистрированных пользователей обеспечивает доступ всех заинтересованных категорий лиц к просмотру и поиску адаптированных ОПОП, просмотру информации о вузе, реализующему данные ОПОП, а также доступ к мультимедийному контенту, описывающему условия обучения по адаптированной ОПОП (Рис. 3).



оператору проекта.

Функции, доступные незарегистрированному пользователю:

- просмотр и поиск адаптированных ОПОП с использованием следующих фильтров: название ОПОП, регион расположения вуза, реализующего ОПОП, нозологическая группа;
- просмотр сведений об ОПОП: УГСН; направление; нозологическая группа, для которой адаптирована данная программа; уровень образования; форма обучения; выпускающая кафедра; заведующий выпускающей кафедры; руководитель программы; описание программы в текстовом виде с использованием мультимедийных элементов в описании; требования к абитуриентам; файл, содержащий описание программы;
- просмотр информационных страниц образовательной платформы;
- отправка сообщений с помощью формы обратной связи оператору проекта;
- просмотр сведений о вузе, реализующем ОПОП: краткое и полное наименование вуза; географическое расположение вуза; общие сведения о вузе; данные о кадровом персонале, курирующем обучение лиц с ОВЗ; условия для включения лиц с ОВЗ в образовательный процесс, имеющиеся в вузе.

Модули платформы работают с заданным набором информационных объектов. Каждый информационный объект представляется отдельной таблицей базы данных.

Программное обеспечение прототипа образовательной платформы представляет собой распределенное веб-приложение, разработанное с применением паттерна проектирования MVC на объектном языке PHP и JavaScript с использованием библиотеки jQuery. Веб-приложение функционирует под управлением веб сервера Apache и СУБД MySQL [1]. Пользовательский интерфейс интерактивной карты России, реализованный в личном кабинете Министерства, реализован с помощью SVG графики, Javascript и HTML 5 [2].

Использование образовательной платформы позволит информировать абитуриентов с ОВЗ о наличии в существующих вузах адаптивных ОПОП, об условиях обучения, а также позволит более гибко управлять реализацией ОПОП в конкретных вузах и

регионах, осуществлять мониторинг и ситуативное реагирование со стороны Министерства образования и науки РФ.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Прохоренок Н.А. HTML, Javascript, PHP и MySQL. Джельтменский набор Web-мастера. – СПб.: БХВ-Петербург, 2008. – 640 с.
2. Дунаев В.В. HTML, скрипты и стили. – СПб.: БХВ-Петербург, 2011. – 816 с.

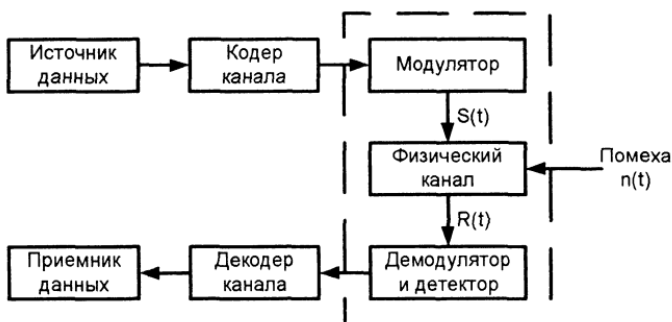
**Н.Н. ГРИНЧЕНКО, Е.А. КЛИМЕНКО**

Рязанский государственный радиотехнический университет

### РЕАЛИЗАЦИЯ МОДЕЛИ СИСТЕМЫ ПЕРЕДАЧИ ЦИФРОВЫХ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ МАССИВНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

*Рассматривается модель системы передачи данных, идеи и разновидности методов помехоустойчивого кодирования. Обосновывается эффективность массивно-параллельных вычислений при моделировании систем передачи цифровых данных.*

Структура типовой системы передачи цифровых данных [1] изображена на рис. 1.



**Рис. 1 – Структурная схема системы передачи цифровой информации.**

Моделируемая система должна имитировать работу каждого компонента данной схемы. Модулятор, физический канал и демодулятор с детектором объединим в канал передачи данных, подверженный

воздействию помех, так как при моделировании на компьютере не имеет смысла переводить цифровой сигнал в аналоговую форму и обратно.

Для моделирования источника данных необходимо использовать генератор случайных чисел, который мог бы выдавать различные последовательности нулей и единиц с равной вероятностью нуля и единицы. Для этого можно использовать генератор с биномиальным распределением или распределением Бернулли. На момент написания статьи модель системы передачи данных реализует алгоритм кодирования Хэмминга и использует генератор случайных чисел с распределением Бернулли.

На физический канал в системе передачи цифровых данных действуют помехи, которые могут приводить к искажению битов исходного сигнала. В результате на выходе канала передачи данных получается измененный сигнал. Для борьбы с искажениями сигнала используются различные методы. Наряду с улучшением физических характеристик канала, широкое распространение получили алгоритмы помехоустойчивого кодирования [2-6], которые позволяют обнаруживать и исправлять возникающие ошибки. Этот способ является более предпочтительным по сравнению с улучшением характеристик канала передачи данных по той причине, что требует меньших затрат на реализацию, так как может быть реализован программно.

В основе методов помехоустойчивого кодирования лежит идея внесения избыточности в исходное сообщение, которая в случае возникновения ошибок позволит обнаружить и, возможно, исправить их. Различные коды имеют различные корректирующие возможности и области применения, поэтому для их оценки нужно использовать модели.

Но моделирование системы передачи цифровых данных является вычислительно-затратной задачей, так как для изучения характеристик конкретного кода требуется сгенерировать и закодировать большое количество данных. Например, если мы хотим оценить вероятность ошибки декодирования 1E-12, нужно смоделировать передачу по каналу связи хотя бы 1E13 бит.

На момент написания статьи модель реализует код Хэмминга [7,8], который является линейным блоковым систематическим кодом с кодовым расстоянием  $d_{\min} = 3$ . Минимальное кодовое расстояние равно минимальному количеству символов в блоке, которое должно измениться, чтобы одно разрешенное сообщение перешло в другое разрешенное сообщение.

Закодированное сообщение получается путем умножения исходного сообщения на порождающую матрицу вида

$$G_{k*n} = [E_{k*k} P_{k*(n-k)}],$$

где  $k$  — количество информационных символов,

$n$  — общее количество символов в закодированном сообщении,

$E$  — единичная матрица размера  $k * k$ ,

$P$  — проверочная матрица размера  $k * (n - k)$ , строки которой состоят из различных комбинаций нулей и единиц, причем в каждой строке должно быть более одной единицы. Каждый столбец матрицы  $P$  показывает, какие биты контролируются проверочным битом, стоящим на соответствующей позиции в получаемом кодовом слове.

То есть процесс кодирования кодом Хэмминга можно записать следующим образом:

$$V_{1*n} = I_{1*k} * G_{k*n},$$

где  $V_{1*n}$  — вектор закодированного сообщения длины  $n$ ,

$I_{1*k}$  — вектор исходного сообщения длины  $k$ .

В основу кода Хэмминга положен принцип синдромного декодирования. Полученное искаженное сообщение умножается на транспонированную проверочную матрицу  $H^T$  следующего вида:

$$H_{(n-k)*k} = [P^T E_{(n-k)*(n-k)}].$$

То есть, обозначив синдром как  $S$ , если было принято искаженное сообщение  $V'$ , имеем:

$$S_{1*(n-k)} = V' * H^T.$$

В случае, когда сообщение было принято без искажений, синдром равен нулю. В противных случаях синдром укажет на позицию ошибки, если произошла однократная ошибка.

Код Хэмминга можно использовать либо только для исправления одиночных ошибок, либо для обнаружения всех двойных ошибок. Также может иметь место ситуация, когда количество ошибок кратно  $d_{\min}$ , тогда одна разрешенная кодовая комбинация переходит в другую разрешенную кодовую комбинацию, и ошибку обнаружить не удастся. То есть данный код позволяет снизить вероятность ошибки в принятом сообщении.

На момент написания статьи модель реализует три версии алгоритма Хэмминга:

- 1) для расчета на одном ядре центрального процессора;
- 2) для расчета на всех доступных ядрах центрального процессора с использованием технологии OpenMP;



3) для расчета на ядрах видеокарты с использованием технологии Cuda [9].

Таблицы 1-3 содержат результаты замеров производительности различных версий алгоритма и позволяют говорить о высокой эффективности массивно-параллельных вычислений.

Максимальное полученное ускорение с использованием технологии Cuda составило 10,9 раза по сравнению с последовательной версией программы, производящей расчеты на одном ядре центрального процессора.

Расчет ускорения производился по следующей формуле:

$$S(n) = \frac{T(1)}{T(n)},$$

где  $T(1)$  – время вычислений на последовательной версии,

$T(n)$  – время вычислений на параллельной версии алгоритма.

**Таблица 1 – Сравнение различных версий алгоритма для кода (7, 4).**

|        | Количество генерируемых сообщений | Вероятность битовой ошибки | Время выполнения, с | Количество обнаруженных ошибок | Ускорение |
|--------|-----------------------------------|----------------------------|---------------------|--------------------------------|-----------|
| CPU    | 1000                              | 0.15                       | 2.225               | 646                            | 1         |
| OpenMP | 1000                              | 0.15                       | 1.526               | 658                            | 1.46      |
| Cuda   | 1000                              | 0.15                       | 0.896               | 689                            | 2.48      |
| CPU    | 10000                             | 0.01                       | 2.645               | 728                            | 1         |
| OpenMP | 10000                             | 0.01                       | 1.429               | 714                            | 1.85      |
| Cuda   | 10000                             | 0.01                       | 0.397               | 666                            | 6.7       |

**Таблица 2 – Сравнение различных версий алгоритма для кода (15, 11).**

|         | Количество генерируемых сообщений | Вероятность битовой ошибки | Время выполнения, с | Количество обнаруженных ошибок | Ускорение |
|---------|-----------------------------------|----------------------------|---------------------|--------------------------------|-----------|
| CPU     | 1000                              | 0.15                       | 2.243               | 858                            | 1         |
| Open MP | 1000                              | 0.15                       | 1.363               | 883                            | 1.65      |
| Cuda    | 1000                              | 0.15                       | 0.226               | 888                            | 9.92      |
| CPU     | 10000                             | 0.01                       | 2.915               | 1440                           | 1         |
| Open MP | 10000                             | 0.01                       | 1.5                 | 1378                           | 1.94      |
| Cuda    | 10000                             | 0.01                       | 0.399               | 1359                           | 7.3       |

**Таблица 3 – Сравнение различных версий алгоритма для кода (63, 57).**

|         | Количество генерируемых сообщений | Вероятность битовой ошибки | Время выполнения, с | Количество обнаруженных ошибок | Ускорение |
|---------|-----------------------------------|----------------------------|---------------------|--------------------------------|-----------|
| CPU     | 1000                              | 0.15                       | 2.474               | 979                            | 1         |
| Open MP | 1000                              | 0.15                       | 1.497               | 986                            | 1.65      |
| Cuda    | 1000                              | 0.15                       | 0.227               | 984                            | 10.9      |
| CPU     | 10000                             | 0.01                       | 4.49                | 4746                           | 1         |
| Open MP | 10000                             | 0.01                       | 2.495               | 4770                           | 1.8       |
| Cuda    | 10000                             | 0.01                       | 0.914               | 4667                           | 4.9       |

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник. М.: Горячая линия-Телеком, 2004. – 126 с.
2. Код Хэмминга. Пример работы алгоритма. [Электронный ресурс] URL: <https://habrahabr.ru/post/140611>.
3. Блейхуг Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. М.: Мир, 1986. – 576 с.

4. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. – М.: Радио и связь, 1987. – 392 с.
5. Березюк Н.Г. Кодирование информации (двоичные коды). Харьков: Издательское объединение «Вища школа», 1978 – 252 с.
6. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. М.: Мир, 1976. – 593 с.
7. Код Хэмминга. [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Код\\_Хэмминга](https://ru.wikipedia.org/wiki/Код_Хэмминга).
8. Hamming code. [Электронный ресурс] URL: [https://en.wikipedia.org/wiki/Hamming\\_code](https://en.wikipedia.org/wiki/Hamming_code).
9. cuRAND :: CUDA Toolkit Documentation. [Электронный ресурс] URL: <http://docs.nvidia.com/cuda/curand/#axzz4U1bvpyCk>.

**А.И. ГУСЕВА**

Рязанский государственный радиотехнический университет

### **ИСПОЛЬЗОВАНИЕ СИСТЕМЫ T-FLEX CAD ПРИ РАЗРАБОТКЕ МОДУЛЯ УПРАВЛЕНИЯ СОЕДИНИТЕЛЬНЫМ УСТРОЙСТВОМ**

*В данной статье рассматривается система T-FLEX CAD. Приведены основные возможности и достоинства программы.*

Сформировавшиеся экономические обстоятельства предъявляют промышленным предприятиям соответствующие современные условия. В ситуации конкурентной борьбы особую важность принимает факт сжатия времени подготовки конструкторско-технологической специализации на производстве. Для решения существующей проблемы необходимо использовать интегрированную систему автоматизированного проектирования (САПР), которая объединяет все нужные подсистемы в одном информационном пространстве. Хорошим примером интегрированной САПР служит программный комплекс T-FLEX CAD/CAM/CAE/CAPP/PDM от компании «Топ Системы».

Рассматриваемая система является совокупность комплекса САПР автоматизирующих некоторые стадии конструкторско-технологической специализации на производстве, это факт показывает, что предприятие имеет право использовать необходимые компоненты системы. Программный комплекс T-FLEX имеет высокую сте-

пень интеграции, что допускает сократить временные потери при передаче сведений из одной подсистемы в другую.

При разработке модуля управления соединительным устройством был использован САПР T-FLEX CAD. На сегодняшний день система проектирования радиоэлектронной аппаратуры T-FLEX CAD представляется достаточно распространённой. T-FLEX CAD - система параметрического автоматизированного проектирования и черчения. При использовании системы возникло впечатление о легкости в использовании программы, графическом удобстве, высокой степени гибкости и возможности реконструкции чертежа в условии сохранения соотношений между элементами.

Хочется обратить внимание на не сложную работу меню пиктограмм, интуитивное общение между пользователем и системой внутри команд, практичную манипуляцию элементами чертежа и библиотеками чертежей, имеется контекстно-зависимая функция помощи, которая облегчает обучение. Буквально любая команда системы создана таким образом, чтобы у пользователя не возникало проблем в работе конструирования.

Средства системы T-FLEX CAD являются высокоэффективными, они предоставляют использование ее для большого объема задач. Основное назначение данной системы, это применение в конструкторской работе, разработке многопланового оборудования и инструментов, также разработка чертежей штампов и пресс-форм, проектирование законченных изделий. Рассматриваемая программа представляет полный комплект возможностей для выполнения конструкторских и технологических чертежей: построение линий всевозможных типов, текстов штриховок, размеров, шероховатостей, специальных символов. Все необходимые возможности для конструирования модуля управления реализовались при помощи данной системы. Все вышеперечисленные средства связаны с параметрами чертежа. Это значит, что реконструкция величин чертежа автоматически приводит к изменению скоординированных элементов построения. Чертежи созданы в соответствии с требованиями ЕСКД, при желании можно в одно мгновение изменить законченный чертеж из одного стандарта в другой.

Программа T-FLEX CAD при разработке изображения применяет определенные типы элементов: элементы построения, элементы изображения, вспомогательные элементы. Основание чертежа осуществляется элементами построения, с ними взаимосвязаны элементы изображения. Такие элементы сопоставляются с тонкими карандашными линиями, которые потом обводятся мягким карандашом, к ним относятся узлы и линии построения. Таким образом, при создании

данных линий построения и узлов, выполняется связь между элементами построения и устанавливается хронология пересчета из расположения при параметрической реконструкции чертежа. Эти линии и узлы мы видим только на экране, они не печатаются на бумагу или кальку.

Размеры, штриховки, тексты и допуски, размещение поверхностей и главное, видимые при печати линии изображения – это все называется элементами изображения. Такие линии являются привязанными к элементам чертежа и при операции реорганизации линий и узлов чертежа, аналогично меняют свое расположение, в этом свойстве лежит основа параметрического чертежа. Такие свойства системы как базы данных, отчеты, переменные примыкают к рядам вспомогательных элементов.

Существует несколько способов выполнения чертежа: параметрический и непараметрический. При формировании чертежей модуля управления я применяла параметрический чертеж. Работа с параметрическими чертежами является главным режимом системы T-FLEX CAD. Выполненный параметрический чертеж имеет возможность стать элементом параметрической библиотеки, чтобы применить его в более сложных чертежах, также имеется функция измены параметров и расположения этого элемента.

Выполнение элементов построения является началом создания параметрического чертежа. Для начала нужно создать базовые линии строения от них создаются новые линии, они могут быть вертикальными или горизонтальными, на их основе выполняются построения окружностей, прямых и др. Таким образом, система, имея интуитивные навыки, может запомнить метод выполнения построений. Для дальнейших построений нужно создать узлы, которые создаются на пересечении прямых. Выполняя эти операции, я создаю параметрические связи элементов чертежа, другими словами задаю параметрический чертеж.

Следующий шаг построения, идущий за нанесением вспомогательных линий, выполнение элементов, с помощью которых создается изображения конструкции. При их начертании программа привязывает их к выполненным на предварительном этапе элементам построения – узлам и линиям. Следующее действие оформление чертежа. Под этим термином скрываются операции нанесения размеров, привязка к линиям и узлам чертежа, выполнение штриховки (там, где она необходима) и ее типа заполнения, ввод текстовой информации. Текст также привязан к элементам построения, делается это не автоматически, а в случае необходимости, чтобы текст изменял свое расположение вместе с кор-

рекции чертежа. На данном этапе работы получен параметрический чертеж, он может быть изменен, к примеру, я могу реконструировать расстояние параллельной линии от базовой, угол наклона, радиусы окружностей. Данная возможность является очень полезной и необходимой, так как в несколько действий за определенное время, получается несколько вариантов одного и того же чертежа.

Линии построения в системе T-FLEX CAD можно создавать при помощи задания явного математического описания, или по-другому говоря, с помощью функции. При задании линии при помощи функции нужно указать тип ее представления, он может быть параметрическим или явным, начало и конец отсчета реконструированного параметра, характеристику вырисованной линии. При разработке чертежей модуля управления функция позволила задать линию построения практически любого вида, данная возможность системы очень удобна, если необходимо создать форму детали, при невозможности использования стандартных функций.

Применение функции возможно в двух режимах: применение имеющейся формулы или задание другой функции. Набор предлагаемых функций находится в файле «function.dat», при необходимости готовую функцию можно изменить частично, либо заменить полностью своей формулой. Полученная функция рассматривается системой как сплайн. Расположение заключительной линии в нужном месте чертежа происходит с помощью системы координат, при выборе двух узлов. Расположение точки 0,0 определяется первым узлом, а вторым ориентация оси абсцисс (X), ось ординат (Y) задается автоматически.

Задание функциональной зависимости и других параметров необходимо чтобы построить функцию. Для задания могут быть использованы следующие формы: явные и параметрические. Явное задание в системе координат ( $Y=f(X)$ ) применяется для выполнения линий параболы. Параметрическое задание в декартовой системе координат ( $X=f(t)$ ) задает прямую ( $Y=f(t)$ ). Функция в полярных координатах ( $P=f(A)$ ) служит для выполнения окружности. Параметрическое задание в полярной системе координат ( $A=f(t)$ ,  $P=f(t)$ ) используется реже, но может быть полезно в некоторых ситуациях, например, при помощи данной функции есть возможность сохранить значение углов и расстояний, определяющих координаты кулачка. Для конструирования модуля управления в рассматриваемой системе мне пригодились явное задание в системе координат ( $Y=f(X)$ ) и параметрическое задание в декартовой системе координат ( $X=f(t)$ ).

В результате построения получается полилиния, начало расчета и завершение определяется согласно значениям применяемого пара-

метра начальному и конечному соответственно. Существует два способа формирования координат точек при построении полилинии: с оптимизацией по кривизне и с постоянным шагом. Первый случай предусматривает, что изменяемый параметр преобразовывается от первого значения до последнего с определенным равномерным шагом. От количества шагов зависит число участков полилинии и качественная форма искривления. Для кривых непростой организации не всегда удобен данный способ, потому что на всех участках, ровных и искривленных плотность разбиения будет одинаковой при условии, что это не всегда нужно. Использование оптимизации по кривизне дает редкое разбиение полилинии на меньшей кривизне и более частое при обратной ситуации. Имеется возможность применить полученные полилинии, как сплайн, мне было необходимо воспользоваться этой, функцией чтобы обрести гладкую кривую, применяя заданное количество определяющих точек. Для выполнения сплайна необходимо включить команду «Создать сплайн по полилинии». Затем задается тип и число сегментов между двух смежных точек сплайна для полилинии, в завершении операции будет получена полилиния на основе сплайна.

В системе T-FLEX CAD чертежи, которые сделанные с применением фрагментов именуют сборочными. Сборочные чертежи не составляют в себе информации фрагментов, а только ссылаются на них. Данное свойство дает возможность хранить чертежи наиболее компактно в памяти запоминающего устройства, это не единственное преимущество. Еще один плюс этой возможности, в том, что сложные чертежи гораздо проще создаются, так как изначально чертятся отдельные элементы сборки, а потом объединяются. В сборочном чертеже модуля управления имеются чертежи деталей, которые повторяются, описанная выше функция очень удобна в данном случае. Необходимо принимать к вниманию, что линии и другие элементы фрагмента не часть сборочного чертежа. Это значит, что на их основании не получится обычным способом, например, задать размеры, для этого в программе имеется специализированный тип узлов – узлы с фрагмента. Создать такой узел возможно при помощи команды Node или автоматизировано при добавлении в чертеж детали.

Для реконструирования расположения элементов сборочного чертежа при изменении все элементы, которые включены в рассматриваемый чертеж, необходимо привязать к остальным элементам. При выполнении данной функции существует 2 способа привязки: векторы привязки и точки привязки. Различие этих методов состоит в хронологии конструирования чертежа-детали. Используя вектор привязки, сначала создается чертеж, а затем определяются векторы привязки, а

при применении второго метода изначально строятся точки привязки, а затем конструируется чертеж. Стоит немного рассказать о способах построения векторов и точек привязки подробнее.

Вектор привязки служит для привязки чертежа, как было уже сказано ранее, чертеж в который он вводится, приобретает статус сборочного. На чертеже допускается до 20 таких векторов, также используемый вектор устанавливает ориентацию детали при вставке в чертеж. Создается он при помощи функции программы FV: Построить вектор привязки. Задать вектор привязки можно двумя типами: единичной точкой или двумя. При задании первым типом соответственно привязка запрашивается единичной точкой, данный фрагмент не может поворачиваться. Если во всплывающем меню команды убрать метку «Использовать только первую точку», будет создан вектор по двум точкам, направление его будет совпадать с осью X чертежа детали. Выбранный по умолчанию вектор, при вставке чертежа детали в сборочный используется как главный вектор привязки, все остальные являются неосновными. Положение фрагмента в сборке определяется первой точкой привязки, а вторая определяет поворот детали. При конструировании чертежей модуля управления применение векторов привязки оказалось очень простым и удобным, я пользовалась привязкой, задаваемой по двум точкам.

В системе T-FLEX CAD имеется еще одна очень удобная функция, чертежи можно дополнить изображениями, которые хранятся в графических форматах: grb, bmf, bmp, wmf, emf. Изображения могут быть вставлены с различным масштабом, под различными углами. Изображения также, как и детали в сборочном чертеже, привязаны к узлам, только имеется одно различие, изображения не имеют переменных и специальных элементов привязки, т.е не являются параметрическими. Механизм привязки в данном примере также отличен от способов, описанных выше для элемента фрагмента.

Применяется два метода привязки к изображению на чертеже, первый это применение двух угловых точек привязки, второй использование одной точки привязки, угла поворота и масштаба. Рассмотрим данные методы подробнее. Принцип привязки изображения к чертежу при помощи точек привязки заключается в следующем, представим, что рисунок вставки как бы ограничен прямоугольником, необходимо задать два противоположных угла этой фигуры на чертеже. Таким образом, имеется возможность изменять расположение и размер изображения на чертеже. Для того чтобы изменять расположение и размер рисунка, необходимо привязать угловые точки к узлам чертежа.



Второй метод осуществляется немного по-другому. В данном способе точка привязки это одна из опорных точек рисунка, которая из опорных точек будет применена, задается параметрами выравнивания в окне диалога параметров. Данный метод также дает возможность задать разворот рисунка, из-за того, что точка привязки привязана к узлу, параметры картинки регулируются при помощи масштабного коэффициента. Вставка изображения в чертеж осуществляется с использованием команды IP, включаемому изображению нужно дать название, которое может быть переменным. При конструировании модуля управления я не использовала вставку картинки в чертеж, так как это не было необходимым, но в данной статье решила рассказать о такой возможности системы T-FLEX CAD, потому что эта функция очень полезна.

Применение переменных в программе T-FLEX CAD позволило приподнять понятие параметризации на наивысший уровень без использования программирования, проводить непростые математические расчеты в чертеже, задавать привязку между элементами. Так что же такое переменная в программе T-FLEX CAD? Это точно такой же элемент как, например, линия построения, но различие есть, заключается оно в характере действия. Если при создании линии построения она сразу проявляется на экране монитора и на ее базе можно продолжить строить чертеж далее, то переменная оказывает действие на чертеж более скрыто. Чтобы увидеть результат действия переменной на чертеж, необходимо установить взаимосвязь между переменной и нужным элементом конструирования.

Присвоение имени и значения необходимо каждой переменной, которые просчитываются в согласованности с математической формулой, также нужно указать комментарий к переменной, который показывает, что фактически она представляет: длину, массу, радиус и т.д. Переменные представляются двух типов: вещественные и текстовые. Вещественные принимают значение числа, текстовые соответственно строку символов.

Математическая формула содержит типовые логические, алгебраические расчеты, условные операции которых обращаются к математическим функциям и функциям T-FLEX CAD. Итогом выражения являются значения, которые соответствуют типу переменной, выражение может содержать операнды (текстовые константы и переменные, вещественные константы и переменные), операции (комбинация операций над этими операндами), функции.

В системы T-FLEX CAD переменные создаются несколькими методами. При помощи редактора переменных, в текстовом редакторе

задавая и редактируя линии строения, при использовании текстовых строк в параметрах элементов, при задаче назначения абсолютно всех вещественных параметров элементов. Применение переменных при конструировании модуля управления, в качестве значения уровня видимости позволило закрыть одной параметрической моделью более широкий набор вариантов.

Использование системы T-FLEX CAD является эффективным методом решения целого ряда задач при создании чертежей. В данной статье было рассказано о многочисленных плюсах данного программного комплекса, о различных способах построения чертежа, использовании допустимых команд для простоты и удобства работы. При конструировании модуля управления соединительным устройством программа T-FLEX CAD оказалась несложна в использовании и практична. Очень удобным свойством программы оказалось создание сборочных параметрических чертежей, применение функций и переменных. Даже у неопытного пользователя не возникнет особых сложностей с данной программой.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Руководство пользователя T-FLEX Parametric cad двухмерное проектирование и черчение. М.: АО «Топ Системы», 1999г. – 428с.

**Д.А. ЕВСТАФЬЕВ**

Рязанский государственный радиотехнический университет

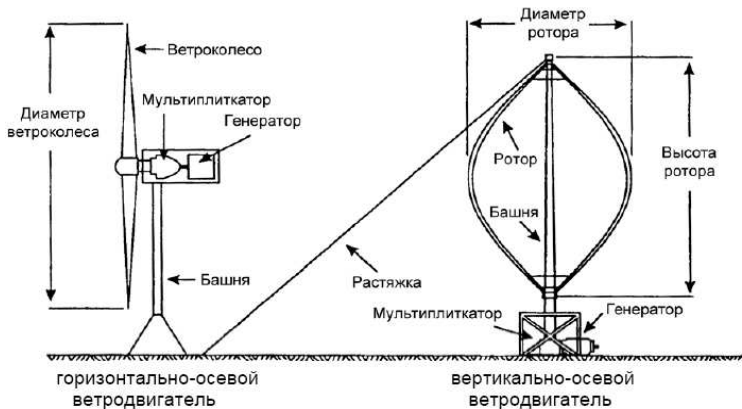
### **КОНСТРУИРОВАНИЕ СИСТЕМЫ ПИТАНИЯ ВЕТРОЭНЕРГЕТИЧЕСКОЙ УСТАНОВКИ**

*В статье представлена информация по конструированию ветроэнергетической установки, проблемах и их возможных решениях при конструировании системы питания ветроэнергетической установки.*

Ресурсный потенциал ветра для выработки энергии бесконечен. А его повсеместная доступность является огромным преимуществом. Получение электроэнергии с помощью ветроэнергетической установки не только позволит в перспективах экономить на электроэнергии, но и сохранить экологию. Но прежде чем энергия ветра сможет принести значительную пользу, должны быть решены многие проблемы, главные из которых: высокая стоимость ветроэнергетических установок, их способность надежно работать в течение долгого периода времени

и обеспечивать бесперебойное электроснабжение. Поэтому, наиболее важной задачей, стоящей сегодня перед ветроэнергетикой, является снижение удельной стоимости электрооборудования ветроэнергетической установки. Одним из таких путей является применение более экономичных структур электрооборудования ветроэнергетической установки (ВЭУ).

Для начала стоит отметить, что в настоящее время применяются две основные конструкции ветроагрегатов: горизонтально-осевые и вертикально-осевые. Отличительная особенность ветроэлектрогенераторов с вертикальной осью вращения состоит в том, что вследствие своей геометрии при любом направлении ветра они находятся в рабочем положении. К тому же, такая схема позволяет установить мультипликатор (редуктор) с генератором внизу башни, что упрощает и удешевляет конструкцию (Рис. 1).



**Рис. 1 – Основные виды ветродвигателей.**

Мультипликатор (редуктор) – промежуточное звено между ветроколесом и электрогенератором, который повышает частоту вращения вала ветроколеса и обеспечивает согласование с оборотами генератора.

Для понятия принципа построения ветроэнергетической установки, представим её схему (Рис. 2).

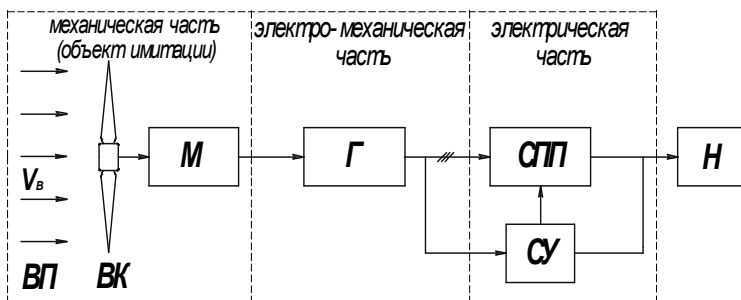


Рис. 2 – Структурная схема ВЭУ.

ВП – ветровой поток; ВК – ветряное колесо; М – мультипликатор; Г – генератор, СПП – силовые полупроводниковые преобразователи; СУ – схема управления; Н – нагрузка.

Рассмотрим более подробно СПП в системе электропривода. Электропривод представляет собой электромеханическую систему, состоящую из взаимодействующих электрических преобразователей, управляющих и информационных устройств и устройств сопряжения с внешними сопредельными электрическими, механическими, управляющими и информационными системами.

Базовым элементом электропривода является электродвигатель, который осуществляет электромеханическое преобразование энергии. В настоящее время электроприводы оснащают различными электронными преобразователями электроэнергии и микропроцессорными системами, что существенно расширяет функционал системы.

Электропривод имеет два канала – силовой и информационный. По первому транспортируется преобразуемая энергия, по второму осуществляется управление потоком энергии, а также сбор и обработка сведений о состоянии и функционировании системы, диагностика ее неисправностей (Рис. 3).

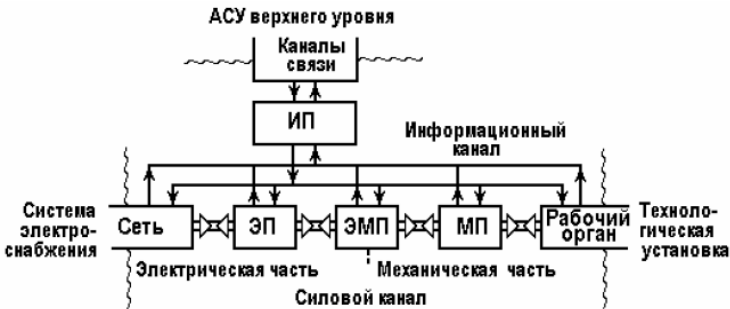


Рис. 3 – Общая структура электропривода.

Силовой канал состоит из двух частей – электрической и механической, которые связаны между собой при помощи электрохимического преобразователя (ЭМП).

В электрическую часть силового канала входит устройство электрического преобразователя ЭП, передающего электрическую энергию от источника к электрохимическому преобразователю ЭМП и обратно. При необходимости, ЭП также может преобразовывать электроэнергию.

Механическая часть состоит из подвижного органа электрохимического преобразователя, механических передач и исполнительного органа установки. Электрохимический преобразователь ЭМП преобразует электрическую энергию в механическую и обратно.

Практически все процессы, связанные с механической энергией и движением, осуществляются электроприводом. Электропривод взаимодействует как с источником электрической энергии или с системой электроснабжения, так и с технологической установкой или машиной, так и с информационной системой более высокого уровня через информационный преобразователь (ИП).

Из всего вышеперечисленного следует, что электродвигатель является главным потребителем электрической энергии. Он потребляет более 60% всей производимой электроэнергии, что делает особенно острой проблему энергосбережения в электроприводе.

В настоящее время с появлением новых технологий электроприводы также совершенствуются. Немаловажную роль также играет их большая популярность, которая с ростом числа ветроэнергетических установок будет только расти, а стоимость – снижаться. Но основной целью неизменно является повышение КПД электродвигателя.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <http://ecoterem.com/netradicionnye-i-vozobnovlyaemye-istochniki-energii/zapasy-energii-vetra-i-vozmozhnosti-ee-ispolzovaniya/>
2. <http://mysagni.ru/fea/energetics/1155-naznachenie-i-klassifikaciya-elektroprivodov.html>
3. <http://msd.com.ua/razdel-energосnabzhenie/ustrojstvo-vetroelektricheskoj-ustanovki/>

**Е.А. ЖДАНКИНА, В.С. РЕШЕТОВА, Е.А. СОФЬИН**

Рязанский государственный радиотехнический университет

## РЕАЛИЗАЦИЯ АЛГОРИТМОВ СЖАТИЯ ИЗОБРАЖЕНИЙ НА БАЗЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ И ИХ СРАВНЕНИЕ

*Реализованы алгоритмы сжатия изображений на базе вейвлет-преобразования. Произведено тестирование и сравнение вейвлетов Хаара и Добеши.*

### 1. Введение

Проблема сжатия изображений привлекает внимание специалистов при решении широкого спектра прикладных задач. На сегодняшний день известно множество алгоритмов, основанных на вейвлет-преобразовании. Наиболее известные – JPEG 2000, DjVu.

Вейвлеты – это обобщенное название семейств математических функций определенной формы, которые локальны во времени и по частоте и в которых все функции получаются из одной базовой (порождающей) посредством ее сдвигов и растяжений по оси времени. Как правило, вейвлет-преобразования (WT) подразделяют на дискретное (DWT) и непрерывное (CWT). Набор вейвлетов может приближать сложный сигнал или изображение точно или с некоторой погрешностью [1-3].

Целью данной работы является реализация, анализ и сравнение базовых алгоритмов на основе вейвлетов Хаара и Добеши.

### 2. Основы вейвлет-преобразования

Вейвлет-преобразование сигнала можно рассматривать как представление сигнала в виде суперпозиции некоторых базисных функций – волновых пакетов. Прототипная волна может быть рассмотрена в качестве импульсной реакции базового фильтра. В этом случае вейвлет-преобразование сведется к совокупности процессов фильтрации и децимации (Рис. 1).

Для фильтрации преобразуемого сигнала используются фильтры нижних и верхних частот, которые разделяют диапазон частот исходного сигнала на две части. Полученные компоненты сигнала имеют в два раза более узкую полосу частотных составляющих. Поэтому, в соответствии с теоремой Котельникова, их можно дискретизировать и сжать. После низкочастотных и высокочастотных фильтров каждый второй отсчет может быть просто исключен, что означает децимацию и прореживание.

На втором этапе преобразования низкочастотная компонента вновь разделяется на низкочастотную и высокочастотную составляющие с помощью аналогичных фильтров. После прореживания возможно повторение этой процедуры.



**Рис. 1 – Схема дискретного вейвлет-преобразования.**

Количество вейвлетов, которые используются при разложении сигнала, определяет уровень декомпозиции сигнала. В качестве нулевого принимают сам сигнал, а уровни декомпозиции образуют ниспадающее вейвлет-дерево. Точность представления сигнала падает по мере перехода на более низкие уровни декомпозиции, однако появляется возможность применения вейвлет-фильтрации сигналов шумов и эффективной компрессии.

При прямом преобразовании происходит разложение произвольного входного сигнала на новый базис в виде совокупности волновых пакетов – вейвлетов [1,2].

### 3. Непрерывное вейвлет-преобразование

Как правило, в основе вейвлет-преобразований лежит использование двух непрерывных, взаимозависимых и интегрируемых по независимой переменной функций: вейвлет-функции  $\psi(t)$  и масштабирующей функции  $\varphi(t)$ .

Вейвлет-функция  $\psi(t)$  имеет нулевое значение интеграла

$$\int_{-\infty}^{\infty} \psi(t) dt = 0.$$

С помощью данной функции, которую часто и называют вейвлетом, выделяются локальные особенности сигнала.

Она порождает детализирующие коэффициенты:

$$d_{j-1} = \int s(t) \bar{\psi}_{j-1}(t) dt.$$

Масштабирующая функция  $\varphi(t)$ , как временная скейлинг-функция с единичным значением интеграла  $\int_{-\infty}^{+\infty} \varphi(t) dt = 1$ , выполняет

грубое приближение (аппроксимацию) сигнала. Она порождает коэффициенты аппроксимации:

$$a_{j-1} = \int s(t) \bar{\varphi}_{j-1}(t) dt.$$

$\varphi(t)$ -функции присущи не всем, а, как правило, только ортогональным вейвлетам.

#### 4. Вейвлет Хаара

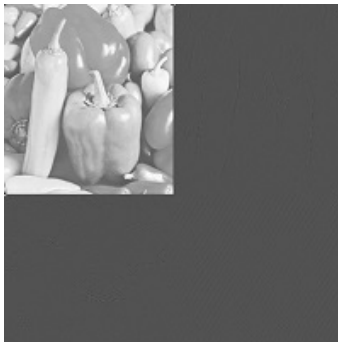
Один из самых известных ортогональных вейвлетов – вейвлет Хаара (Haar). Его порождающая скейлинг-функция задается следующим выражением:

$$\varphi(t) = \begin{cases} 1, & 0 < t < 1 \\ 0, & t < 0, t > 1 \end{cases},$$

$\psi(t)$ -функция принимает вид прямоугольных импульсов меандра:

$$\psi(t) = \begin{cases} 1, & 0 < t < 0.5 \\ -1, & 0.5 < t < 1 \end{cases}.$$





**Рис. 2 – Изображение, к которому применено прямое вейвлет-преобразование Хаара.**

### **5. Вейвлет Добеши**

Вейвлеты Добеши не имеют аналитического выражения и определяются только фильтрами. Скейлинг-функции и вейвлеты Добеши – это непрерывные функции, которые не тождественны нулю и недифференцируемы на конечном отрезке.

Вид скейлинг-функции и вейвлета Добеши вычисляются при помощи метода итераций, а в практических приложениях используются только вейвлет-коэффициенты  $h_k$  и  $g_k$  без вычисления конкретной формы вейвлетов.

Область задания вейвлетов Добеши шире, чем вейвлетов Хаара, но при этом они обеспечивают большее количество малозначимых коэффициентов разложения и, при отбрасывании последних, более сильное сжатие данных [4,5].

### **6. Реализация и результаты**

Были реализованы два алгоритма вейвлет-преобразований: простейший вейвлет Хаара, а также биортогональный вейвлет Козна-Добеши-Фово. Среда разработки – Spider 3.0.0, язык программирования Python 3.5.2. Использовались библиотеки PIL (Python Imaging Library), Numpy, Matplotlib, Math.

Для тестирования работы программ проводилось преобразование изображения размером 512 x 512 px в градациях серого.

Во всех опытах исходное изображение подвергалось прямому преобразованию, после чего обнулялись значения пикселей, которые оказались меньше заданного порога. После чего промежуточный вариант сжимался при помощи архиватора 7zip. Далее промежуточное изо-

бражение подвергалось обратному преобразованию (восстанавливалось).

Для оценки степени достоверности восстановления изображения использован критерий отношения пикового значения сигнала к шуму (*PSNR* – peak signal-to-noise-ratio):

$$PSNR = -10 \log_{10} \left( \frac{MSE}{S^2} \right),$$

где *MSE* – среднеквадратичная ошибка (mean square error), равная сумме квадратов разностей между оригиналом и восстановленным изображением, а *S* – максимальное значение пикселя:

$$MSE = \frac{1}{MN} \sum_{m=0}^M \sum_{n=0}^N |x(m, n) - \hat{x}(m, n)|^2.$$

Под коэффициентом сжатия будем понимать отношение размера архива с исходным изображением к размеру архива с преобразованным изображением.

Размер архива с исходным изображением – 176 Кб. Результаты расчетов представлены в таблицах 1 и 2.

**Таблица 1 – Вейвлет Коэна-Добеши-Фово.**

| Порядок | Порог обнуления | Размер архива с преобразованным изображением, Кб | Коэффициент сжатия | <i>MSE</i> | <i>PSNR</i> |
|---------|-----------------|--|--------------------|------------|-------------|
| 2       | 0               | 109  | 1,61               | 1.38 e-31  | 356.73      |
| 2       | 0,01            | 71,3   | 2,47               | 9.64 e-05  | 88.29       |
| 2       | 0,03            | 29,7   | 5,93               | 0.000344   | 82.76       |
| 2       | 0,04            | 25,4   | 6,929              | 0.00042    | 81.87       |
| 2       | 0,05            | 23,5   | 7,48               | 4.96 e-4   | 81.17       |
| 4       | 0               | 102  | 1,71               | 3.26 e-31  | 353.00      |
| 4       | 0,01            | 62,2   | 2,82               | 1.80 e-4   | 85.56       |
| 4       | 0,03            | 17,6   | 10                 | 0.00095    | 78.35       |
| 4       | 0,05            | 10,1   | 17,43              | 1.80 e-3   | 75.57       |
| 8       | 0               | 129  | 1,36               | 1.09 e-30  | 347, 00     |
| 8       | 0,01            | 65,6   | 2,68               | 2.99 e-4   | 83.38       |
| 8       | 0,03            | 17,3   | 10,17              | 0.00297    | 73.41       |
| 8       | 0,05            | 9,23   | 19,06              | 9.07 e-3   | 68.55       |

Таблица 2 – Вейвлет Хаара.

| Порог обнуления | Размер архива с преобразованным изображением, Кб | Коэффициент сжатия | <i>MSE</i> | <i>PSNR</i> |
|-----------------|--|--------------------|------------|-------------|
| 0               | 125  | 1,41               | 8.73 e-32  | 310.59      |
| 0,01            | 125  | 1,41               | 1.14 e-5   | 49.44       |
| 0,05            | 125  | 1,41               | 1.80 e-4   | 37.43       |
| 0,07            | 125  | 1,41               | 2.23 e-4   | 36.52       |
| 0,3             | 125  | 1,41               | 1.13 e-3   | 29.47       |

## 7. Выводы

На основании проделанных тестов сформулированы следующие выводы:

- 1) Даже преобразование без обнуления значений пикселей, яркость которых ниже заданного порога, ведет к улучшению компрессии;
- 2) Обнуляя коэффициенты можно добиться значительного сжатия при применении последующей архивации (или иного алгоритма сжатия). Но следует отметить, что существует порог, после которого увеличение сжатия становится незначительным, в то время как качество изображения существенно ухудшается;
- 3) Для вейвлета Коэна-Добеши-Фово увеличение порядка ведет к повышению уровня сжатия, но сопровождается потерей качества;
- 4) Для вейвлета Хаара не удалось подобрать коэффициент обнуления, который бы значительно улучшал компрессию при дальнейшей архивации. Повышение коэффициента приводит к ухудшению качества, но при гораздо больших значениях, чем для вейвлета Коэна-Добеши-Фово.

Вейвлет-преобразование можно применять в качестве подготовительного этапа при компрессии сигнала, при этом возникает проблема подбора коэффициента обнуления и порядка вейвлета. Кроме того, в дальнейшем целесообразно проанализировать варианты сжатия, опирающиеся на свойства проделанных преобразований.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Уэлстид С.. Фракталы и вейвлеты для сжатия изображений в действии. Учебное пособие. – М.: Издательство «Триумф», 2003, с. 77 – 119.

2. Воробьев В. И., Грибунин В. Г. Теория и практика вейвлет-преобразований. ВУС, 2003, с. 1 – 204.

3. John Kominek. Advances in Fractal Compression for Multimedia Application. Department of Computer Science, University of Waterloo, Waterloo, Ontario.

4. Rafael C. Gonzalez, Richard E. Woods. Digital Image Procession Third Edition.

5. Давыдов А.В. Вейвлетные преобразования сигналов. Курс лекций – Екатеринбург: УГГУ, ИГиГ, кафедра геоинформатики, 2005.

**В.А. ЖУРАВЛЕВ, В.И. ОРЕШКОВ**

Рязанский государственный радиотехнический университет

## **ИННОВАЦИОННЫЙ ИНЖИНИРИНГ ДЛЯ ОБЪЕКТОВ ИНФРАСТРУКТУРЫ**

*Рассматриваются перспективные направления в области проектирования на основе баз знаний.*

Благодаря техническому прогрессу и использованию бурильных машин в строительстве, количество туннелей в городе растет, а вместе с этим увеличивается возможная длина и глубина объектов, что позволяет называть перевозки посредством метро наиболее успешным решением транспортного вопроса. Несмотря на все плюсы, проектирование нового объекта является сложной задачей. Такие технические аспекты как диаметр туннелей, а также командная работа инженеров варьируются от проекта к проекту. Во время проектирования частый обмен информацией между специализированными командами осуществляется в основном за счет чертежей. Данная технология не способна установить связь между информационными элементами, которые они представляют. Таким образом, и инженеры, и проектировщики должны быть уведомлены обо всех изменениях, появляющихся в проекте, и вносить их в геометрическую модель вручную. Данный процесс требует большого расхода времени и подвержен ошибкам. В результате ограничений в чертежных системах наблюдается постоянный рост использования систем параметрического компьютерного конструирования (CAD) в архитектуре, инженерных системах и строительстве. Несмотря на стабильность данной модели и ускорение темпов проектирования, которые предлагают CAD, процесс проектирования остается рутинной работой. Инженеры и эксперты инвестируют 10% своего

времени в административные обязанности, 70% на рутинные задания и только 20% на эффективную работу. Это растягивает процесс проектирования и затрудняет креативный поиск инновационных решений. Большинство знаний инженеров, применяемых в дизайне объектов инфраструктуры метро, основывается на инженерных правилах. Понимание и внедрение их абстрактных знаний в САД систему не является простой задачей. Одно из таких правил входит в расчеты выравнивания объекта инфраструктуры. Вместо работы с 3D пространственной кривой выравнивания инженеры предпочитают работать с двумя 2D образами кривых, горизонтальным и вертикальным выравниванием, которые делают возможным обхождение сложных задач и различных ограничений, напрямую связанных с проектированием выравниваний. Большое количество времени тратится на интерпретацию инженерных правил и перевод в параметрическую 3D модель.

Системы автоматизированного проектирования стали стандартным инструментом практически во всех промышленных областях. В 80х годах прошлого века, системы автоматизированного проектирования сосредотачивались на простом геометрическом образовании и манипуляции, а не поддержке автоматического образования геометрических моделей на основе накопленных знаний. В ответ на эту технологическую проблему, в конце 80-х годов крупные корпорации, такие Airbus и Jaguar начали разрабатывать методологии КВЕ. Определяемые как перекресток основных технологий, таких как искусственный интеллект (ИИ), система автоматизированного проектирования (САПР) и объектно-ориентированное программирование (ООП). Подходы КВЕ захватили и повторно использовали знания инженеров применяемые в процессе проектирования, сократили рутинную работу и, следовательно, увеличили время для разработки более творческих решений. Основная архитектура КВЕ, определенная LaRossa (2012), состоит из трех модулей, которые анализируют конкретные функциональные требования, такие как размер, стоимость и производительность, которые предлагают использовать инкапсулированные знания, содержащиеся в правилах, кодах и таблицах дизайнера, и которые возвращают оптимальные инженерные конструкции в виде рисунков или 3D-моделей.

В настоящее время системы КВЕ имеют устойчивое присутствие в механической и авиационной промышленности. Интерес в системах знаний возрос настолько быстро в последние два десятилетия, что утверждают, будто в течение следующего десятилетия методы КВЕ будут иметь такое же значение, что САД / САМ / САЕ системы была в 1990-е годы.

Рассматривая различные подходы к внедрению систем КВЕ, Реноуег и др. (2000), классифицировали их по методу решения использования и типа запрашиваемого пользовательского интерфейса. Согласно этой классификации, генеративные методы КВЕ подходят для управления, без какого-либо другого воздействия, создания или модификации окончательной геометрии на основе заранее заданных ограничений и пользовательского ввода информации. Эта генеративная способность имеет большое значение при проектировании инфраструктуры, потому что мы стремимся отделить концептуальный проект от трудоемкой задачи геометрической генерации. Уже в конце 1980-х годов Адели и Баласубраманиам - известные эксперты в области дизайна того времени описали первую порождающую систему КВЕ, разработанную для расчета мостовых балок. Позже, Гардан и Гардан (2003) сообщили об инновационном подходе, основанном на плане, где задачи внедрения и применения ставились отдельно на основе проверки пользователя - мета описание для инженера-программиста и описание знаний для пользователя САПР. Тем не менее, ни один из предыдущих примеров, не связал систему КВЕ с параметрическими системами автоматизированного проектирования. Один из первых подходов, которые связали систему КВЕ с параметрической системы САПР, открыли Мен и Хань (2001). Их подход представил новую методологию КВЕ, которая была в состоянии модифицировать параметрические узлы, основанные на концепции модуля проекта, функциональной части и функциональной особенности.

Пока концепции дизайна устройства и функциональной части содержат знания, которые могут управлять узлами и деталями, функциональная особенность остается основным геометрическим элементом, принадлежащим каждой системе САД. Несмотря на то, что функциональные возможности могут быть повторно использоваться от одного проекта к другому, главным недостатком такого подхода остается тот факт, что каждый блок конструкции или функциональная часть подходили конкретному узлу или его части и этот факт не мог быть полностью интегрирован, пока модель не была закончена.

Дизайн объектов инфраструктуры сильно регулируется методическими указаниями, нормами и национальными стандартами. Понимание и внедрение базовых абстрактных сведений в систему САПР не такая простая задача. Кроме того, специалисты должны интерпретировать требования этих инженерных правил, их соответствие конкретным требованиям, и выражать свои решения в соответствующей модели продукта. В заключение, проектное задание отнимает много времени и тормозит творческий поиск инновационных решений.

Одним из существенных преимуществ моделирования этих технических правил в параметрической системе САД является то, что конечная (результативная) геометрия производится после того как примененное правило заключено в нескольких конструкторских операциях, которые могут быть легко распознаны и изолированы в конструкторской истории. Кроме того, интерпретация таких инженерных правил может быть легко собрана в системе КВЕ таким образом, что пользователю необходимо только предоставить входную информацию и затем покинуть систему, чтобы вернуться к новой или обновленной процедурной геометрии. В частности, каждая логическая модель резюмирует (содержит) другое инженерное правило и способна интерпретировать входные параметры и генерировать соответствующий набор процедурных операций. Такой подход также позволяет инженерам и специалистам изменять целые (законченные) секции объекта инфраструктуры простой модификацией одного параметра, и практически мгновенно визуализировать полученную геометрию.

Одно ограничение существующих генеративных систем КВЕ является фактом того, что сведения входят (объединяются) в процесс проектирования после создания первой геометрической модели. Это ограничение заставляет разработчиков создавать большие библиотеки моделей шаблонов, которые могут ответить на конкретные вопросы проектирования. Несмотря на то, что это ограничение трудно преодолеть, дизайн на основе инженерных правил дает нам возможность создавать КВЕ структуры заранее. Таким образом, мы определяем концепцию логических моделей в качестве компонента базовой архитектуры КВЕ, описываемой LaRossa (LaRossa, 2012), где вход является абстрактным сведением, с которым инженер должен иметь дело, а выход набора процедурных операций необходим, чтобы смоделировать конечную геометрию с помощью параметрической системы САПР.

По мере того как конструкция объекта инфраструктуры развивается, сведения, необходимые для моделирования также становятся все более сложными. Для того, чтобы избежать повторения в определении знаний и необходимости разработки логических моделей, которые описывают все возможные сценарии, несколько логических моделей могут быть горизонтально соединены, чтобы достичь более сложного моделирования. Простой пример двух логических моделей, которые могут быть связаны для достижения более сложного моделирования, можно найти в подгонке и осевых кривых кольцевого туннеля. С одной стороны, логическая модель выравнивания содержит информацию, необходимую для создания пространственной кривой - описываемая кривая на середине пути между двумя рельсами - на основе

горизонтальных и вертикальных рядов. С другой стороны, логическая модель отвечает за оси туннеля - используемая для направления бурильной машины для работы при проходке туннелей (ТВМ) - создает вторую пространственную кривую, которая смещается из кривой выравнивания по вертикальным и горизонтальным параметрам. Поскольку ось туннеля основывается на кривой, описываемой выравниванием, обе модели могут быть связаны, чтобы избежать избыточности и уменьшить логические сложности. Следовательно, изменение в модели выравнивания будет обновлять геометрию выравнивания и кривой оси - даже тогда, когда модель оси туннеля остается неизменной. Такое поведение достигается за счет горизонтальных элементов соединения, которые позволяют логическим моделям взаимодействовать друг с другом. Горизонтальные разъемы обеспечивают связь типа «один-многим», которая фиксирует изменения и логическое содержание от соответствующей модели к связанным моделям. Хотя рабочий процесс в логической модели, в основном, происходит сверху вниз, от логического блока к геометрическому слою, несколько логических моделей могут быть соединены горизонтально, чтобы создать более сложный сценарий моделирования. Кроме того, горизонтальные связи предоставляют два различных типа связи между логическими моделями: активные и пассивные соединения. Активное соединение позволяет всем связанным моделям быть проинформированными при возникновении обновления в соответствующей логической модели.

Литература и практический опыт показывают, что пользователи САПР предпочитают работать с функциональными системами, чем с прямым манипулированием рельефа поверхности. В параметрических системах САПР процедурные операции, которые не основаны на эскизах, как правило, следуют этому функциональному подходу, при котором пользователи только представляют семантическую информацию о рельефе, который они хотят создать или изменить, в то время как система САД манипулирует рельефом и устанавливает необходимые зависимости. Успешное принятие методологии КВЕ обусловливается ее интеграцией в рабочий процесс инженера. Такая интеграция должна дополнительно гарантировать согласованность модели, обеспечивающей сосуществование традиционных и КВЕ подходов поколения геометрии.



## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Статья Assisted design of railway runnels based on logic models and advanced procedural geometry dependencies by Amor. R

**В.О. КИСЕЛЕВА**

Рязанский государственный радиотехнический университет

### **РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ, ИССЛЕДУЮЩЕГО СЕТЕВЫЕ ХАРАКТЕРИСТИКИ В SDN СЕТЯХ**

*Разработан программный модуль, измеряющий задержку во время пересылки пакетов от одного хоста к другому в SDN сетях.*

В современном мире бизнес в сфере информационных технологий предъявляет все большие требования к гибкости и масштабируемости компьютерных сетей.

Традиционные архитектуры/дизайны сетей становятся неэффективны в динамических средах. Традиционные сети прежде всего статичны и не соответствуют быстрой динамике развития современного IT бизнеса. Программно-конфигурируемые сети (Software Defined Networking) позволят исправить данные недостатки.

В SDN уровни управления сетью и передачи данных разделяются за счет переноса функций управления (маршрутизаторами, коммутаторами и т. п.) в приложения, работающие на отдельном сервере (контроллере). Такие технологии позволяют повысить эффективность сетевого оборудования, снизить затраты на эксплуатацию сетей, повысить безопасность и предоставить пользователям возможность программно создавать новые сервисы и оперативно загружать их в сетевое оборудование.

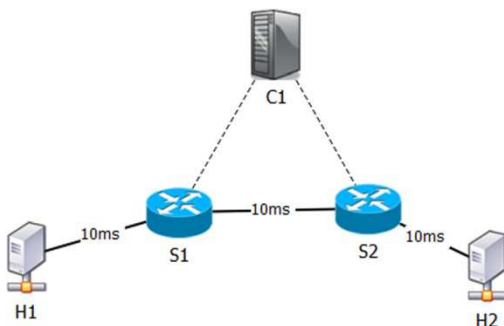
Контроллер – это выделенный сервер, на котором работает специальное программное обеспечение, состоящие из сетевой операционной системы и сетевых управляющих приложений. Отвечает за обеспечение взаимодействия приложения-коммутатор. Предоставляет важные сервисы (например, построение топологии, мониторинг хостов). Различают такие разновидности контроллеров: Nox, Pox, MUL, Ruy, Beacon, OpenDaylight, Floodlight, Maestro, McNettle, Flower, Runos – разные языки программирования от Python до Haskell, Erlang.

В данных контроллерах используются таблицы потоков, которые описывают, как наиболее эффективно доставить пакет от отправителя

к пункту назначения. В качестве правил выбора оптимального маршрута для сети, часто используют следующие алгоритмы маршрутизации: Дейкстры, Йена и Беллмана-Форда. Для алгоритма Дейкстры, маршрут создаётся исходя из параметров сети (задержка, пропускная способность, процент потерь пакетов). Так как данные параметры постоянно изменяются, для наиболее эффективного выбора маршрута необходимо постоянно измерять данные сетевые характеристики.

Модуль, рассматриваемый в данной статье, измеряет задержку между двумя каналами. Так как для измерения других параметров сети существует утилита `iperf`.

Задержка передачи определяется как задержка между моментом поступления пакета на вход какого-либо сетевого устройства или части сети и моментом появления его на выходе этого устройства. Этот параметр характеризует только сетевые этапы обработки данных, без учета задержек обработки компьютерами сети.



**Рис. 1 – Топология для измерения задержки с помощью модуля.**

В данном модуле задержка измеряется между двумя свитчами с учетом контроллера (Рис. 1). Поток пакетов от одного свитча отправляется к другому, а затем сразу же передается обратно. Данная операция необходима для того, чтобы измерять время каждый раз в момент отправки потока. Разница между данными измерениями и время ожидания ответа от контроллера будут составлять итоговую задержку. Параметр в сети измеряется каждую секунду. Результат измерения для представленной топологии изображен на рис. 2.

Для того чтобы составить таблицу потоков для всей сети необходимо с помощью данного модуля последовательно отправлять пакеты между свитчами с помощью утилиты `ping`.

В результате, применяя модуль и утилиту `iperf` можно максимально оптимизировать маршрут исходя из параметров сети в данный момент времени.

```
start_time: 1.38408400288e+12
INFO:core:POX 0.1.0 (beta) is up.
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
ConnectionUp: 00-00-00-00-00-01
INFO:openflow.of_01:[00-01-00-00-00-01 2] connected
ConnectionUp: 00-01-00-00-00-01
INFO:recoco:Task <Timer/tid4> scheduled multiple times
delay: 10.7114257812 ms
delay: 9.14892578125 ms
delay: 8.83288574219 ms
delay: 10.3375244141 ms
delay: 9.14904785156 ms
delay: 9.47253417969 ms
delay: 10.0002441406 ms
INFO:openflow.of_01:[00-00-00-00-00-01 1] closed
ConnectionDown: 00-00-00-00-00-01
INFO:openflow.of_01:[00-01-00-00-00-01 2] closed
ConnectionDown: 00-01-00-00-00-01
```

**Рис. 2 – Результат измерения задержки на представленной топологии.**

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. OpenFlow Tutorial. [http://archive.openflow.org/wk/index.php/OpenFlow\\_Tutorial](http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial)
2. Руслан Смелянский. Программно-конфигурируемые сети. <https://www.osp.ru/os/2012/09/13032491/>
3. Measure the latency. <http://csie.nqu.edu.tw/smallko/sdn/latency.htm>

**А.А. КОСЫРЕВ**

Рязанский государственный радиотехнический университет

### **АВТОМАТИЧЕСКАЯ КОРРЕКТИРОВКА ПОГРЕШНОСТИ ИЗМЕРИТЕЛЬНОГО УСТРОЙСТВА**

*Рассматриваются показатели методов автоматической корректировки погрешностей, нахождение влияющих условий, определение погрешности и формирование корректирующего сигнала.*

Главным характерным показателем методов автоматической корректировки погрешностей является обеспечение ими близости ста-

тической настоящей функции преобразования измерительного устройства (ИУ) номинальной характеристике преобразования путем изменения настоящей функции преобразования ИУ под влиянием корректирующего сигнала. Таким образом, основной операцией процесса автоматической корректировки погрешности ИУ является определение этой погрешности и формирования соответствующего корректирующего сигнала.

- нахождение влияющих условий  $z_1, z_2, \dots, z_n$  и расчета погрешности по известной зависимости для используемого ИУ  $\Delta = F(z_1, z_2, \dots, z_n)$ ;

- определение погрешности ИУ, подведенной к выходу,  $-\Delta_y$ ;

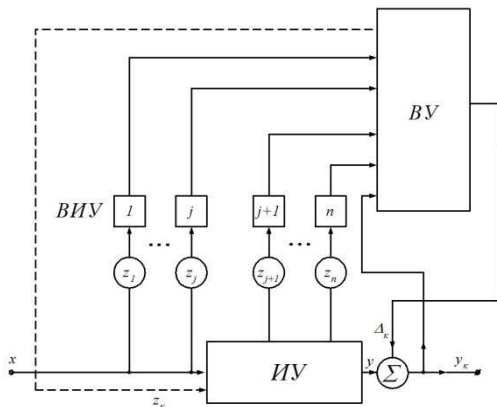
- определение погрешности ИУ, подведенной ко входу,  $-\Delta_x$ .

Для определения погрешности с целью ее автоматической корректировки необходимо знать отношение погрешности от влияющих на нее условий и, поскольку в процессе использования ИУ они изменяются, необходимо периодически измерять эти условия.

Первая операция называется *методом вспомогательных измерений* и показана на рис. 1.

На основании отклонений  $\Delta z_j = z_j - z_{nj}$  влияющих величин от номинальных значений вычислительное устройство (ВУ) по известным функциям определяет погрешность  $\Delta_k$ , которая суммируется с выходным сигналом, или вырабатывает корректирующий сигнал  $z_k$ , воздействующий на параметры настоящей функции преобразования основного измерительного устройства (ИУ).

ВУ по данным вспомогательных измерений значений  $\Delta z_j$ , известной номинальной функции  $f_n(x)$  и по функциям влияния  $\psi_j$  вычисляет номинальное значение входного сигнала  $x_n = f_n^{-1}(y)$ , где  $f_n^{-1}(y)$  – функция, обратная номинальной, и поправку  $\Delta_k = F(z_1, z_2, \dots, z_m)$  либо корректирующий сигнал  $z_k$ . Для простоты и большей наглядности будем в дальнейшем рассматривать введение поправки. В точке суммирования  $\sum$  образуется разность.



**Рис. 1 – Метод вспомогательных измерений.**

Некорректированное значение выходного сигнала:

$$y = f(x) - f_n(x) + \Delta y,$$

где  $\Delta y = F(z_1, z_2, \dots, z_n)$  – погрешность на выходе до коррекции.

В точке суммирования  $\Sigma$  образуется разность:

$$y_k = y - \Delta_k = f_n(x) + \Delta y - \Delta_k = f_n(x) + \Delta_{ky};$$

где  $\Delta_{ky} = \Delta y - \Delta_k$  – погрешность после коррекции, приведенная к выходу. Погрешность после коррекции, приведенная ко входу,

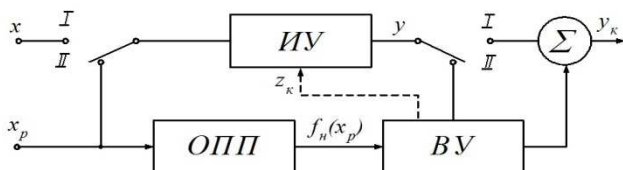
$$\Delta_{kx} = f_n^{-1}(y_k) - x.$$

Вторая процедура выработки корректирующего сигнала носит название *метод образцовых сигналов* и иллюстрируется упрощенной структурной схемой (Рис. 2).

Метод широко применяется для коррекции погрешности нуля измерительных устройств ( $x_p = 0$ ). Применение метода для коррекции мультипликативных погрешностей аналоговых устройств ограничивается трудностью создания точного образцового преобразователя ОПП при ( $x_p = 0$ ).

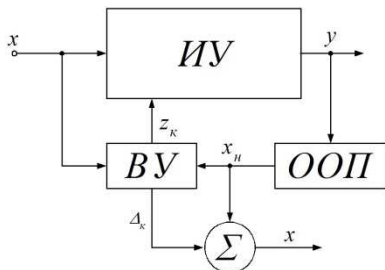
Таким образом, можно отметить основные характерные особенности метода: в состав системы коррекции входит образцовый прямой преобразователь, от точности которого существенно зависит точность коррекции; корректируется суммарная погрешность устройства при тех условиях, при которых подключается образцовый сигнал; погрешность корректируется лишь в дискретных точках (при  $x = x_p$ ), а в промежуточных точках определяется с помощью ВУ путем интерпо-

лирования, поэтому для перекрытия всей шкалы прибора необходимо иметь несколько образцовых значений  $x_p$ ; коррекция осуществляется не непрерывно, а периодически.



**Рис. 2 – Метод образцовых сигналов.**

Третья процедура – *метод обратного преобразования* (Рис. 3) содержит образцовый обратный преобразователь (ООП) для выработки номинального значения входной величины  $x_n = f_n^{-1}(y)$  и получения разности  $\Delta_x = x - x_n$  приведенной ко входу.



**Рис. 3 – Метод обратного преобразования.**

Как видно из структурной схемы, при этом методе автоматическая коррекция погрешности осуществляется непрерывно. Он во многом сходен с методом введения отрицательной обратной связи, но в отличие от последнего позволяет корректировать суммарную погрешность как мультипликативную, так и аддитивную. При этом чувствительность преобразователя не уменьшается.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Земельман М.А. Автоматическая коррекция погрешностей измерительных устройств. М.: Издательство стандартов, 1972.
2. Сергеев А.Г., Крохин В.В. Метрология. М.: Логос, 2001.
3. Садовский Г.А. Основы теории погрешностей измерительных устройств. Рязань, 1981.
4. Пиотровский Я. Теория измерений для инженеров. М.: Мир, 1989.

**М.А. ЛУНЯКОВА, М.А. БАКУЛЕВА**

Рязанский государственный радиотехнический университет  
Рязанский государственный медицинский университет им. акад. И.П. Павлова

### МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РЕЗУЛЬТАТОВ ПЕРВИЧНОЙ ДИАГНОСТИКИ ДЛЯ ЭКСПЕРТНОЙ СИСТЕМЫ ВРАЧА- ГЕМАТОЛОГА

*Рассматриваются проблемы разработки ЭС врача-гематолога. Предлагается математическая модель интерпретации результатов общего анализа крови пациента.*

При разработке экспертных систем (ЭС) в различных предметных областях возникает необходимость интерпретации первичных данных, заносимых из базы данных (БД), для дальнейшего использования машиной вывода. При этом хорошо формализуемыми являются модели данных, подчиненные законам математической логики. Однако, большинство данных врачебной практики являются слабоструктурированными по сути и достаточно трудной аппроксимируются на множестве значений  $(0, 1)$ . В статье предлагается подход к построению математической модели исходных данных, полученных в результате первичной медицинской диагностики (общего анализа крови), для дальнейшей интерпретации и построения совещательного компонента ЭС.

Исходными данными для консультации врача – гематолога являются результаты общего анализа крови (ОАК), бланк которого приведен на рис. 1.

Для типографии!  
при изготовлении документа  
формат А5

Код формы по ОКУД \_\_\_\_\_  
учреждения по ОКПО \_\_\_\_\_

Министерство здравоохранения СССР  
Медицинская документация  
Форма N 224/у

Утверждена Минздравом СССР  
04.10.80 г. N 1030

наименование учреждения \_\_\_\_\_  
Лаборатория \_\_\_\_\_

## АНАЛИЗ КРОВИ N \_\_\_\_\_

«...» \_\_\_\_\_ 20.. г.  
дата взятия биоматериала

Фамилия, И., О. \_\_\_\_\_

Возраст \_\_\_\_\_

Учреждение \_\_\_\_\_ Отделение \_\_\_\_\_ палата \_\_\_\_\_

Участок \_\_\_\_\_ медицинская карта N \_\_\_\_\_

|   |   | Ре-<br>зуль-<br>тат | Норма           |                       |                                 |                             |
|---|---|---------------------|-----------------|-----------------------|---------------------------------|-----------------------------|
|   |   |                     | Единицы СИ      |                       | Единицы, подлежа-<br>щие замене |                             |
| Гемоглобин  | м |                     | 130,0-<br>160,0 | г/л                   | 13,0-<br>16,0                   | г %                         |
|   | ж |                     | 120,0-<br>140,0 |                       | 12,0-<br>14,0                   |                             |
| Эритроциты  | м |                     | 4,0-<br>5,0     | · 10<br>в ст.12<br>/л | 4,0-<br>5,0                     | млн. в 1<br>куб.мм<br>(мкл) |
|   | ж |                     | 3,9-<br>4,7     |                       | 3,9-<br>4,7                     |                             |
| Цветовой по-<br>казатель  |   |                     | 0,85-<br>1,05   |                       | 0,85-<br>1,05                   |                             |
| Среднее со-<br>держание ге-<br>моглобина<br>в 1 эритро-<br>ците |   |                     | 30-35           | пг                    | 30-35                           | пг                          |
| Ретикулоциты  |   |                     | 2-10            | %                     | 2-10                            | %                           |
| Тромбоциты  |   |                     | 180,0-<br>320,0 | · 10<br>в ст.9<br>/л  | 180,0-<br>320,0                 | тыс. в 1<br>куб.мм<br>(мкл) |
| Лейкоциты   |   |                     | 4,0-<br>9,0     | · 10<br>в ст.9<br>/л  | 4,0-<br>9,0                     | тыс. в 1<br>куб.мм<br>(мкл) |

Рис. 1 – Бланк общего анализа крови.

Заметим, что в самом благоприятном случае, все показатели должны входить в диапазон определенных значений (норма).

В данной статье рассматривается врачебная практика педиатра-гематолога, для взрослых пациентов диапазон нормальных значений может варьироваться в зависимости от пола. Например, для показателя нормального уровня тромбоцитов предусмотрен диапазон (180-320), для показателя гемоглобина (110-130) – и.т.д. Таким образом, факт попадания результатов анализа конкретного пациента в утвержденный Минздравом диапазон нормальных значений по каждому исследуемому показателю можно рассматривает как булеву переменную, где нулевое значение будет соответствовать отступлению от нормы.



В результате, бланк, отображенный на рис. 1 можно преобразовать в таблицу истинности (Таблица 1).

**Таблица 1 – Преобразование результатов ОАК в таблицу истинности.**

| Показатель          | Результат | Норма     | Значение истинности |
|---------------------|-----------|-----------|---------------------|
| Гемоглобин          | 112       | 100-120   | <b>1</b>            |
| Эритроциты          | 4,2       | 4-5       | <b>1</b>            |
| Цветовой показатель | 0,9       | 0,85-1,05 | <b>1</b>            |
| Тромбоциты          | 109       | 150-320   | <b>0</b>            |
| Лейкоциты           | 7,7       | 4-9       | <b>1</b>            |
| Эозинофилы          | 3         | 0-5       | <b>1</b>            |
| Базофилы            | 0,5       | 0-1       | <b>1</b>            |
| Лимфоциты           | 30        | 18-40     | <b>1</b>            |

Заметим, что полученное в третьем столбце таблицы значение может быть интерпретировано как булева функция, размерность которой обусловлена числом исследуемых показателей ОАК и вычисляется по формуле:  $2^{\lfloor \log_2 n \rfloor}$ , где  $n$ -количество исследуемых показателей (Таблица 2).

**Таблица 2 – Таблица истинности для полученной функции.**

| $x_1$    | $x_2$    | $x_3$    | $f(x_1, x_2, x_3)$ |
|----------|----------|----------|--------------------|
| <b>0</b> | <b>0</b> | <b>0</b> | <b>1</b>           |
| <b>0</b> | <b>0</b> | <b>1</b> | <b>1</b>           |
| <b>0</b> | <b>1</b> | <b>0</b> | <b>1</b>           |
| <b>0</b> | <b>1</b> | <b>1</b> | <b>0</b>           |
| <b>1</b> | <b>0</b> | <b>0</b> | <b>1</b>           |
| <b>1</b> | <b>0</b> | <b>1</b> | <b>1</b>           |
| <b>1</b> | <b>1</b> | <b>0</b> | <b>1</b>           |
| <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b>           |

Таким образом, в данном примере получена функция от трех переменных  $f(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$ . Переведя эту функцию в десятичный эквивалент, получим значение:  $f(x_1, x_2, x_3)|_{10} = 111$ . Заметим, что в идеальном варианте полученное значение будет соответствовать номеру  $2^{\lfloor \log_2 n \rfloor} - 1$  где  $n$ -количество исследуемых показателей.

И в этом случае дополнительная консультация врача гематолога не требуется.

В случае, когда полученные показания ниже этого значения, необходимо проводить дополнительные исследования. Например, в рассмотренном случае (низкий уровень тромбоцитов) пациенту необходимо провести следующие исследования крови: развернутая коагулограмма, волчаночный антикоагулянт, антитела к кардиолипину, гликопротеину, протромбину (маркеры АФС), антитела к нативной ДНК, маркеры гепатитов В и С [1,2].

На сегодняшний день использование современных информационных технологий, включающих ЭС различного уровня «интеллектуальности» не имеет широкого распространения в практике врача-гематолога. Поэтому провести необходимое дообследование возможно только после назначения врача, при этом необходимо ждать результатов первичных анализов (ОАК), консультирование и дальнейшее дообследование. Все эти процедуры могут занимать порядка 3-5 недель, при этом в случае острого заболевания подобное промедление может быть угрозой жизни. Если система самостоятельно сможет интерпретировать полученные результаты и рекомендовать дополнительное исследование биоматериала (которое проводится аппаратно и, очевидно, достаточно оперативно, что также положительно влияет на информативность исследований), то на консультацию к гематологу пациент попадет с более подробными результатами. Поэтому разработка математической модели для интерпретации первичных данных ОАК является актуальной. Разработанная модель хорошо структурирована и может быть использована машиной вывода для дальнейшей аналитической обработки клинических данных [3].

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Румянцев А.Г., Масчан А.А., Сметанина Н.С., Масчан М.А., Лукина Е.А., Лунякова М.А. Федеральные клинические рекомендации по диагностике и лечению анемии Даймонда-Блекфена у детей. Москва, 2015.

2. Румянцев А.Г., Масчан А.А., Демихов В.Г., Морщакова Е.Ф., Лунякова М.А., Инякова Н.В. Федеральные клинические рекомендации по диагностике и лечению анемии хронических болезней у детей. Москва, 2015.

3. Бакулев А.В., Бакулева М.А. Построение ассоциативных правил на основе дифференцирования графовой модели анализируемой выборки // Вестник Рязанского государственного радиотехнического университета. 2013. № 46-2. С. 86-89.

**П.П. МАРКИНА**

Рязанский государственный радиотехнический университет

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ БЕЗОПАСНОСТИ HRU, СОДЕРЖАЩАЯ ПРОВЕРКУ ЕЕ СОСТОЯНИЯ БЕЗОПАСНОСТИ**

*Рассматривается программная модель безопасности HRU, а также алгоритм, который имеет возможность проверки данной модели на наличие утечки прав (состояния безопасности).*

Дискреционная политика безопасности – политика безопасности, основанная на дискреционном управлении доступом, которое определяется двумя свойствами: все субъекты и объекты идентифицированы, права доступа субъектов на объекты системы определяются на основании некоторого внешнего по отношению к системе правила. Основным элементом систем дискреционного разграничения доступа является матрица доступов.

Модель HRU используется для анализа систем защиты, реализующих дискреционную политику безопасности.

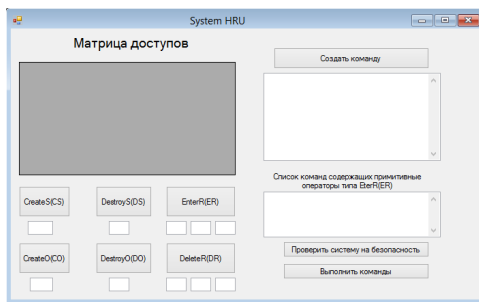
В рамках программной реализации такой модели примем следующие необходимые условности.

Элементами модели будут:

- $S$  – множество возможных субъектов;
- $O$  – множество возможных объектов (напомним, что  $S \subset O$ );
- $R = \{r1, \dots, rn\}$  – конечное множество прав доступа;
- $M$  – матрица прав доступа, описывающая текущие права доступа субъектов к объектам;

Функционирование системы рассматривается только с точки зрения изменений в матрице доступов.

Далее представлено главное окно программы.



**Рис. 1 – Главное окно программы.**

Данная программа реализует модель безопасности Харрисона-Руззо-Ульмана(HRU) для анализа системы защиты. Программа должна задавать дискреционную модель разграничения доступа, иметь возможность выполнения команд и автоматически их анализировать на предмет безопасности (утечки прав).

Описание команд:

1.Внести право  $r$  в  $M[s,o]$  ( $s \in S, o \in O$ ) – добавление субъекту  $s$  права  $r$  по отношению к объекту  $o$ . Команда EnterR(ER).

2.Удалить право  $r$  из  $M[s,o]$  ( $s \in S, o \in O$ ) – удаление у субъекта  $s$  права  $r$  по отношению к объекту  $o$ . Если удаляемое право отсутствовало в ячейке, то состояние системы в результате выполнения данной команды никак не изменяется. Команда DeleteR(DR)

3.Создать субъект  $s$  ( $s \notin S$ ) – создание нового субъекта  $s$ . При создании субъекта в матрицу  $M$  добавляются строка и столбец. Команда CreateS(CS).

4.Уничтожить субъект  $s$  ( $s \in S$ ) – удаление существующего субъекта  $s$ . Команда DestroyS(DS).

5.Создать объект  $o$  ( $o \notin O$ ) – создание нового объекта  $o$ . При добавлении объекта в матрице доступа создаётся новый столбец. Команда CreateO(CO).

6.Уничтожить объект  $o$  ( $o \in O \setminus S$ ) – удаление существующего объекта  $o$ . Команда DestroyO(DO).

Начальное состояние  $Q_0$  считается безопасным по отношению к праву  $r \in R$ , если невозможен переход в такое состояние  $Q$ , в котором возможна утечка права  $r$ . Утечка может произойти при передаче права доступа.

Алгоритм проверки безопасности.

Система находится в начальном состоянии. Первая команда  $Q_0$  создает 3 субъекта с названиями  $S_1, S_2, S_3, S_4$ . Команда  $Q_0: CS[s1]; CS[s2]; CS[s3]; CS[s4]$ ; Команда, включающая 3 примитивных оператора «Создать субъект»,  $CS[S]$  - примитивный оператор create subject.

| Субъекты/Объекты | s1 | s2 | s3 | s4 |  |  |
|------------------|----|----|----|----|--|--|
| s1               |    |    |    |    |  |  |
| s2               |    |    |    |    |  |  |
| s3               |    |    |    |    |  |  |
| s4               |    |    |    |    |  |  |
| *                |    |    |    |    |  |  |

Рис. 2 – Состояние системы после выполнения команды  $Q_0$ .

Вторая команда Q1 вносит право w субъекта S1 на субъект S2. Команда Q1:ER[w; s1;s3]; Команда, включающая один примитивный оператор «Внести право», ER [w; s1, s3]- примитивный оператор enter rule.

| Субъекты/Объекты | s1 | s2 | s3 | s4 |
|------------------|----|----|----|----|
| s1               |    |    | w  |    |
| s2               |    |    |    |    |
| s3               |    |    |    |    |
| s4               |    |    |    |    |

Рис. 3 – Состояние системы после выполнения команды Q1.

Третья команда Q2 вносит право субъекта S1 на субъект S4. Команда Q2:ER [a; s1; s4];

| Субъекты/Объекты | s1 | s2 | s3 | s4 |
|------------------|----|----|----|----|
| s1               |    |    | w  | a  |
| s2               |    |    |    |    |
| s3               |    |    |    |    |
| s4               |    |    |    |    |

Рис. 4 – Состояние системы после выполнения команды Q2.

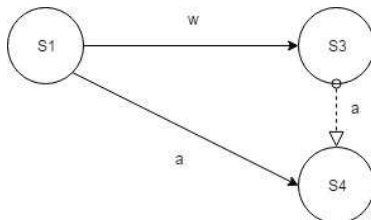


Рис. 5 –Граф состояния системы.

Построив граф, можно увидеть, что субъект S3 через право w может получить право a на субъект S4.

Вывод: рассмотрев конечное состояние матрицы, было выявлено, что данное состояние системы не безопасно, возможна утечка права a.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Девянин П.Н. «Модели безопасности компьютерных систем: Учебное пособия для студентов высших учебных заведений» / Издательский центр «Академия» 2005.
2. Цирюлов В.Л. «Основы информационной безопасности автоматизированных систем» / Издательство «Феникс» 2008.

С.А. МЕЛЕХИН, Д.С. ЮНЮШКИН

Рязанский государственный радиотехнический университет

## ПРИМЕНЕНИЕ И УПРАВЛЕНИЕ ТЕХНОЛОГИЕЙ АФАР НА БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТАХ

*Рассматривается активная фазированная антенная решетка (АФАР) при определенных технических требованиях.*

Разработка многофункциональных бортовых комплексов с активными фазированными антенными решетками (АФАР), использующими единую систему излучения в интересах нескольких радиотехнических систем - важная проблема, стоящая на повестке дня. Решение такой задачи требует разработки принципов построения и технических решений фазированных антенных решеток (ФАР) с широкой полосой частот (до октавы и более), которая перекрывает частотные полосы используемых систем и обеспечивает требуемые характеристики согласования и направленности в широком диапазоне углов сканирования  $\pm(45-60)^\circ$ . При этом, к бортовым ФАР, предъявляются жесткие массогабаритные требования, необходимость учета влияния элементов конструкции и условий эксплуатации.

Требования: тип антенны - АФАР. Р\*G – 2.5\*107 Вт. Ширина ДН в плоскости азимута  $2.5^\circ \pm 0.5^\circ$ , в плоскости угла места  $3^\circ \pm 0.5^\circ$ . УБЛ на передачу минус 21 дБ, на прием минус 25 дБ. Рабочий диапазон частот 8-10 ГГц. Диаметр апертуры АФАР не более 1 м.

В качестве элементарного излучателя, являющегося основой для построения апертуры проектируемой АФАР, выбрана рупорная антенна, представляющая собой открытый конец волновода. Выбор излучателя такого типа обеспечивает требования по частотному диапазону, КПД и излучаемой мощности. Апертура АФАР представляет собой 1500 излучателей, расположенных в узлах гексагональной сетки. Такое расположение излучателей позволяет уместить в апертуре АФАР большее (по сравнению с линейным) количество излучающих элементов при неизменных размерах антенны.

Для подтверждения правильности теоретических расчетов, проводится моделирование энергетических и оптических характеристик АФАР. Строятся ДН в плоскости азимута и угла места для равномерного амплитудного распределения, и максимальных углов отклонения. Для обеспечения требований к уровню боковых лепестков и требований к ширине луча в режимах приема и передачи, для каждого режима выбирается амплитудное распределение (ступенчатое для режима пе-

редачи, косинус на пьедестале для приемного режима), для которого построены ДН. В разрабатываемой модели учитывается не идеальность ФАР в апертуре АФАР, которая присутствует в реальных образцах, обусловленная технологическим разбросом параметров узлов, из которых построена АФАР, а также погрешностью измерений приборов, используемых при ее настройке. Учитывая то, что одним из важных параметров, предъявляемых к АФАР, является ее надежность (сохранение характеристик при появлении неисправных каналов) в модель АФАР введено 10% неисправных каналов, и проведено моделирование ДН и оценка энергетического потенциала.

За счет введения избыточности излучателей в апертуру АФАР и создания спадающего к краям для приемного режима, и ступенчатого для передающего режима амплитудного распределения, она сохраняет свои параметры в заданных пределах при заданных условиях моделирования.

Для управления АФАР разрабатывается схема управления. Элементарной ячейкой, на основе которых строятся все современные АФАР, являются submodule СВЧ и ВУМ включающие в себя все элементы управления (фазовращатели, аттенюаторы, согласующие усилители, МШУ, переключатели), модуляции и усиления СВЧ сигнала, которые управляются с помощью специализированных драйверов. Таким образом, управление АФАР, сводится к передаче необходимого управляющего кода к драйверу каждого канала АФАР, который выставит необходимый режим работы (прием передача), установит необходимую фазу и амплитуду в канале.

Учитывая вышеизложенные особенности, в качестве построения схемы управления, выбрана командная система управления АФАР, представляющая собой блок управления АФАР, который по командам из БЦВМ формирует управляющие слова для каждого канала, и через согласующее устройство передает их в устройства управления каналами (УУК). Получив управляющее слово, УУК формируют необходимые коды управления для драйверов каналов.

Разрабатываемая схема управления, может использоваться для АФАР с различной геометрией апертуры и различным количеством каналов, путем компоновки УУК в апертуре АФАР и корректировки ПО ПЛИС. Для построенной схемы управления рассчитана временная диаграмма работы, определен объем данных передаваемых из блока управления АФАР в УУК, выбран интерфейс передачи данных и определены электронные компоненты для построения УУК.

Для обеспечения работоспособности УУК, описываются процессы настройки АФАР, на основе которых формируются характеристики

управления (коды управления амплитудой и фазой каналов), которые записываются в flash-память УУК.

Разрабатываемая схема управления АФАР обладает большой гибкостью и потенциалом к модернизации. Так на пример для увеличения скорости фазирования (для АФАР с большим числом каналов) возможно распараллелить вычисление управляющих слов в блоке управления АФАР и передавать их параллельно в УУК (увеличив число согласующих устройств). Также возможно введение дополнительных типов лучей или амплитудных распределений путем записи во flash-память управляющих характеристик.

Подводя итоги моделирования АФАР для беспилотных летательных аппаратов видно, что АФАР является более эффективной антенной системой, по сравнению с другими антеннами, позволяющей осуществлять быстрый обзор пространства, путем электронного сканирования, разрешение АФАР по угловым координатам превышает аналогичные зеркальные антенны. При этом существенно повышается надежность работы АФАР по причине отсутствия большого количества механических деталей и механизмов. В конструкции АФАР можно размещать очень большое число маломощных генераторов. В результате при больших значениях суммарной излучаемой мощности существенно снижается вероятность пробоя воздуха, и так же снижается потребление энергии АФАР, что не маловажно в условиях ограниченности энергоресурсов летательного аппарата.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Воскресенский Д.И., Канащенкова А.И. Активные фазированные антенные решётки. Радиотехника, 2004.
2. Самойленко В.И., Шишов Ю.А. Управление фазированными антенными решётками. Радио и связь, 1983.
3. Сазонов Д.М. Антенны и устройства СВЧ. М.: Высш. шк. 1988.



**А.С. НАЙДЕНОВ**

Рязанский государственный радиотехнический университет

**ПРИМЕНЕНИЕ СЭМПЛИНГА В УСЛОВИЯХ  
НЕСБАЛАНСИРОВАННОСТИ КЛАССОВ**

*Рассматривается проблема несбалансированных классов в задаче классификации, и применение сэмплинга в качестве ее решения. Описывается работа различных методов сэмплинга в случае бинарной классификации.*

В процессе решения прикладных задач с помощью методов машинного обучения эффективность модели во многом зависит от качества предоставленных данных. Нередко в обучающем наборе данных присутствует сильная несбалансированность классов (imbalanced classes), т.е. когда классы в выборке представлены неравномерно (обычно, при соотношении классов более чем 10:1). При этом меньший класс называют миноритарным (минорным), а больший — мажоритарным (доминирующим). Проблема встречается в задачах самых разных областей: диагностика редких заболеваний, обнаружение мошенничества, кредитный скоринг, при этом издержки ошибок классификации могут существенно различаться.

При обучении модели на сильно несбалансированных данных алгоритм классификации может посчитать примеры интересующего миноритарного класса шумом и пропустить их. Это может привести к следующим проблемам:

1. Высокое смещение (high bias), т.е. модель будет прогнозировать смещенные оценки в сторону мажоритарного класса.

2. Завышенные некоторые оценки качества модели: например, точность (accuracy), ROC-кривая (ROC AUC).

3. Слабая значимость признаков, т.е. статистические методы не позволяют отвергнуть гипотезу о том, что коэффициенты при значимых (экспертно) признаках равны 0.

Можно выделить три основных подхода к решению указанной проблемы.

Первый подход связан с изменением распределения классов в выборке. Эта задача решается с помощью определенного отбора, т.е. сэмплинга (sampling), примеров из выборки таким образом, чтобы соотношение числа примеров в классах соответствовало заданным требованиям.

Ко второму подходу относятся все методы распределения весов. Этот принцип основан на присвоении каждому классу числового параметра - веса, который соответствует штрафу за неверную классифи-

кацию. Вес определяется экспертной оценкой либо исходя из распределения классов в обучающей выборке.

В третьем подходе примеры миноритарного класса считаются исключительными, и классификация строится в рамках задачи обнаружения аномалий.

На практике часто используется идея сэмплинга ввиду ее простоты и разнообразия. В этом случае применяют либо добавление в обучающую выборку примеров миноритарного класса (*over-sampling*) или удаление из неё примеров мажоритарного класса (*under-sampling*) с целью достичь сбалансированного распределения. Рассмотрим основные алгоритмы сэмплинга.

Удаление примеров мажоритарного класса (*Undersampling*):

1. Случайный отбор (*Randomized undersampling*). Из мажоритарного класса удаляются случайно выбранные примеры. Достоинствами такого подхода являются его наглядность и простота реализации.

2. Односторонний сэмплинг (*One-Sided undersampling*). Главная идея заключается в нахождении и последующем удалении из набора данных таких примеров мажоритарного класса, которые «зашумляют» выборку. Для решения этой задачи могут использоваться следующие методы:

- Поиск связей Томека (*Tomek links*). Цепью Томека является пара наиболее близких примеров из разных классов. В этом случае, удалению подлежат все примеры мажоритарного класса, которые включены в связи Томека.

- Правило сосредоточенного ближайшего соседа (*Condensed Nearest Neighbor Rule*). Здесь удаляются примеры мажоритарного класса, которые образуют «сгустки» или кластеры, наиболее отдаленные от миноритарных примеров (как правило, это примеры, правильно классифицируемые алгоритмом  $k$  ближайших соседей).

3. Ансамбли моделей (*Model ensembles*). Построение композиции классификаторов на сформированных сбалансированных подмножествах исходной выборки. Здесь можно выделить две отличающихся идеи.

- Объединение моделей (*Stacking*). Каждая модель обучается на индивидуальной сбалансированной выборке, включающей часть мажоритарных примеров и все миноритарные примеры. Далее все модели объединяются для финального результата.

- Бустинг (*Boosting*). Последовательное построение классификаторов. Каждая модель обучается на индивидуальной сбалансированной выборке, созданной на основе всей выборки за исключением тех примеров, которые были правильно распознаны предыдущими моделями.

Генерация примеров миноритарного класса (Oversampling):

1. Случайное дублирование (Randomized oversampling). Добавление в обучающую выборку случайно выбранных примеров из миноритарного класса. Отдельно можно выделить бутстрэп, основанный на многократной генерации выборок с заданным распределением классов на базе имеющейся выборки.

2. Алгоритм SMOTE. Генерация искусственных примеров, которые «похожи» на имеющиеся в миноритарном классе, но при этом не являются дубликатами. Алгоритм позволяет регулировать степень сходства новых примеров. Данный метод чаще всего применяется на практике.

3. Дублирование примеров по кластерам (Cluster-based oversampling). Все примеры каждого класса (с размерами  $N_1 > N_2$ ) независимо разделяются на  $k_1$  и  $k_2$  кластеров соответственно (например, с помощью алгоритма k-means). Далее все кластеры мажоритарного класса балансируются (чтобы иметь приблизительно одинаковый размер) любым методом сэмпинга, а все кластеры миноритарного класса подвергаются процедуре дублирования (oversampling) таким образом, чтобы число примеров в каждом кластере было равно  $N_1 / k_2$ .

Применение сэмпинга имеет свои недостатки. Во-первых, дублирование или генерация новых примеров может привести к переобучению алгоритма, а также увеличению времени построения модели. Во-вторых, удаление примеров может привести к потере важной информации в классе. Тем более, некоторые алгоритмы могут эффективно обучаться и на несбалансированных данных

Тем не менее, на практике сэмпинг часто используется, в том числе и в задаче классификации. В большинстве случаев применение сэмпинга позволяет наглядным образом и проще улучшить качество модели.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Н. Хе, Е.А. Garcia «Learning from Imbalanced Data» // IEEE Transactions on Knowledge and Data Engineering. - 2009. - Vol. 21. P. 1263-1284.

2. М. Gao, X. Hong, S. Chen, C. Harris, E. Khalaf, «PDFOS: PDF estimation based over-sampling for imbalanced two-class problems» // Neurocomputing. - Vol. 138. – P. 248-259.

3. Паклин Н.Б., Уланов С.В., Царьков С.В. «Построение классификаторов на несбалансированных выборках на примере кредитного скоринга» // «Искусственный интеллект». - 2010. 528-534 с.

**А. З. НГУЕН, А. П. ШИБАНОВ**

Рязанский государственный радиотехнический университет

## **ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ РАЗБИЕНИЯ ПРИ РАЗРЕШЕНИИ КОНФЛИКТОВ В СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ С МНОЖЕСТВЕННЫМ ДОСТУПОМ**

*В данной статье рассматриваются алгоритмы разбиения при разрешении конфликтов в среде с общим доступом.*

**Введение.** В последние десятилетия отмечается тенденция активного роста числа систем передачи информации, построенных на основе каналов множественного доступа, таких как радиоканалы и спутниковые каналы связи. Среди методов управления доступом большого числа абонентов к общему каналу особое место занимают алгоритм разбиения в системах множественного доступа с разрешением конфликтов. При достаточно низкой интенсивности входного потока сообщений к абонентам конфликты возникают редко, и задержка сообщения оказывается существенно меньше, чем при использовании других методов множественного доступа.

**Целью работы** является исследование алгоритмов разбиения, имеющих существенное значение для разрешения конфликтов в системах множественного доступа.

**Теоретическая часть.** Алгоритмы разбиения – метод разрешения конфликтов, которые обеспечивают устойчивость без применения каких-либо сложных процедур оценки и к тому же увеличивают диапазон доступных скоростей. При использовании алгоритмов разбиения множество, вступивших в конфликт узлов, разбивается на подмножества, причем узлы одного из них передают в следующем окне. Если конфликт не разрешился, то производится дальнейшее разбиение на подмножества. В дальнейшем при изучении этих алгоритмов мы имеем в виду модель с синхронным каналом, пуассоновским поступающим потоком, конфликтами или успешным приемом, быстрой обратной связью с сигналами 0,1,е, повторной передачей при попадании в конфликт и бесконечным числом узлов[1].

Существует три основных алгоритма разбиения:

1. Алгоритм разбиения с древовидной структурой.
2. Алгоритм разбиения с передачей в порядке поступления (ППП).
3. Алгоритм разбиения с передачей в обратном порядке (ПОП).

*Алгоритм разбиения с древовидной структурой* – первый алгоритм по моменту появления алгоритмов разбиения. Когда возникает конфликт, на пример в  $k$ -м окне, узлы, не участвующие в конфликте, переходят в состояние ожидания и узлы, вступившие в конфликт, разбиваются на два подмножества с использованием подбрасывания симметричной монеты. Первое подмножество узлов передает в окне  $k+1$  и, если это окно пустое или в нем была успешная передача, то второе подмножество передает в окне  $k+2$ . В противном случае, когда еще один конфликт возникает в окне  $k+1$ , первое из двух подмножеств разбивается еще раз и второе подмножество узлов ожидает разрешения этого конфликта.

Описанный порядок передачи легко организовать с помощью стека. Когда возникает конфликт, образовавшее его подмножество узлов разбивается, и каждое получающееся подмножество поступает в стек; затем верхнее подмножество стека выбирается из стека и передается. Когда пакет вступает в конфликт, в счетчик записывается 0 или 1 в зависимости от того, в какое подмножество попадает пакет. Когда счетчик переходит в состояние 0, пакет передается, а если счетчик в ненулевом состоянии, его значение увеличивается на 1 после каждого конфликта и уменьшается на 1 после каждой успешной передачи или пустого окна. Период разрешения конфликта (ПРК) заканчивается по определению в момент окончания успешной передачи или в конце пустого окна, причем когда в стеке не остается ячеек с задолженными пакетами.

Если предыдущему ПРК потребовалось большое число окон, что маловероятно, то новых ожидающих пакетов будет много, они образуют конфликт и будут продолжать образовывать конфликт до тех пор, пока подмножества не станут достаточно малыми в этом новом ПРК. Решение этой проблемы следующее: в конце ПРК множество узлов с новыми пакетами сразу разбивается на  $j$  подмножеств, где  $j$  выбивается таким, чтобы среднее число пакетов в подмножестве было немного больше 1 (немного больше из-за того, что после конфликта скорость передачи временно повышается). Эти новые подмножества помещаются затем в стек, и начинается новый ПРК.

Максимальная скорость передачи, которую обеспечивает этот алгоритм при оптимальном выборе  $j$  как функции среднего числа ожидающих пакетов, равна 0,43 пакет за окно.

*В алгоритме разбиения с ППП* множество, вступивших в конфликт пакетов разбивается на два подмножества в порядке моментов поступления. Применяя этот подход, получаем, что каждое подмножество состоит из всех пакетов, которые поступили в течение некоторого

заданного интервала времени; при возникновении конфликта этот интервал разбивается на два более коротких интервала. Передовая в первую очередь пакеты с более раннего интервала, алгоритм в итоге будет передавать успешно пакеты в порядке их поступления.

В каждый целочисленный момент времени  $k$  в окне  $k$  должно передаваться множество всех пакетов, прибывших во время некоторого более раннего интервала, например от момента  $T(k)$  до  $T(k)+\alpha(k)$  ( $\alpha(k)$ - назначенный интервал для окна  $k$ ). Назначенный интервал разбивается на два равных подынтервала и подынтервал первой очереди  $L$  становится назначенным интервалом для окна  $k+1$ . Таким образом,  $T(k+1)=T(k)$ , а  $\alpha(k+1)=\alpha(k)/2$ . Когда окно пустое, то подынтервал второй очереди  $R$  содержит 2 или более пакетов и поэтому он сразу разбивается, причем интервал  $RL$  становится назначенным интервалом для окна  $k+2$ . Таким образом,  $T(k+2)=T(k+1)+\alpha(k+1)$ , а  $\alpha(k+2)=\alpha(k+1)/2$ . В итоге успешные передачи происходят в окнах  $k+2$  и  $k+3$ , тогда завершая ПРК.

Можно описать алгоритм, который строго выполняется каждым узлом сети. Алгоритм задает назначенный интервал ( т.е.  $T(k)$  и  $\alpha(k)$  и статус ( $\sigma = L$  или  $\sigma = R$ ) для окна  $k$  на основе сигнала обратной связи, назначенного интервала и статуса, относящихся к окну  $k-1$ .

Если сигнал обратной связи равен  $e$ , то:

$$\begin{aligned} T(k) &= T(k-1), \\ \alpha(k) &= \alpha(k-1)/2, \\ \sigma(k) &= L. \end{aligned}$$

Если сигнал обратной связи равен 1 и  $\sigma(k-1) = L$ , то:

$$\begin{aligned} T(k) &= T(k-1) + \alpha(k-1), \\ \alpha(k) &= \alpha(k-1)/2, \\ \sigma(k) &= R. \end{aligned}$$

Если сигнал обратной связи равен 0 и  $\sigma(k-1) = L$ , то:

$$\begin{aligned} T(k) &= T(k-1) + \alpha(k-1), \\ \alpha(k) &= \alpha(k-1), \\ \sigma(k) &= L. \end{aligned}$$

Если сигнал обратной связи равен 0 или 1 и  $\sigma(k-1) = R$ , то:

$$\begin{aligned} T(k) &= T(k-1) + \alpha(k-1), \\ \alpha(k) &= \min[\alpha_0, k - T(k)], \\ \sigma(k) &= R. \end{aligned}$$

Алгоритм разбиения с ППП требует, чтобы все узлы постоянно следили за обратной связью канала. Разработанная модификация позволяет узлам следить за обратной связью только после того, как они

получат пакет для передачи. Идея состоит в том, что пакет в основном передается в соответствии с дисциплиной *передачи в обратном порядке (ПОП)*; значит, самым последним поступившим пакетам нет необходимости знать длину очереди, так как они имеют самый высокий приоритет передачи. Новые поступившие пакеты находятся в состоянии предварительного ожидания до тех пор, пока они не получат количество информации обратной связи, достаточное для обнаружения конца ПРК; затем они присоединяются к ожидающему множеству узлов. Окончание ПРК можно определить по сигналу обратной связи 1, за которым следует сигнал обратной связи 0 или 1.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бертсекас Д., Галлагер Р. Сети передачи данных: Пер. с англ. –М.: Мир, 1989. -544с.
2. Руденков Н.А., Долинер Л.И. Основы сетевых технологий: Учебник для вузов. Екатеринбург: Изд-во Уральского. Федерального ун-та, 2011. – 300 с.

### Н.З. НГУЕН, Ю.А. КОПЕЙКИН

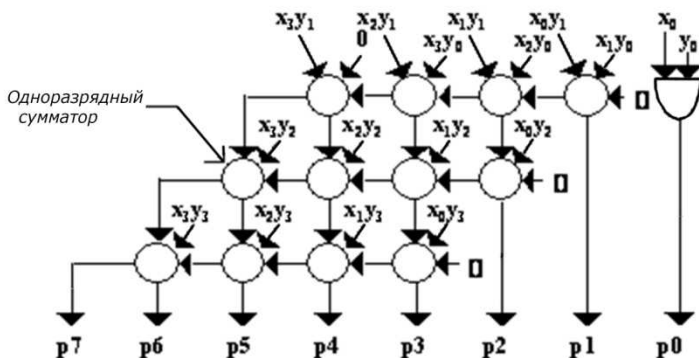
Рязанский государственный радиотехнический университет

### ПОСТРОЕНИЕ БЫСТРОДЕЙСТВУЮЩИХ ПЕРЕМНОЖИТЕЛЕЙ НА БАЗЕ ПЛИС

*Рассматривается построение быстродействующих перемножителей на базе ПЛИС применением алгоритма Бута.*

Реализация операции умножения аппаратными методами всегда являлась сложной задачей при разработке высокопроизводительных вычислителей. Аппаратная реализация алгоритма умножения в первую очередь предназначена для получения максимального быстродействия выполнения этой операции в устройстве. На ПЛИС можно разрабатывать умножители с быстродействием более 100МГц и располагать более 100 модулей умножителей в одном кристалле.

Для операндов небольшой разрядности (4 и менее) наиболее результативна структура простого матричного суммирования (Рис.1). Она реализует параллельный умножитель как массив одноразрядных сумматоров, соединенных локальными межсоединениями, при этом общее число сумматоров напрямую определяется разрядностью множимого и множителя.



**Рис. 1 – Структура матричного умножителя с последовательным распространением переноса.**

Так, полный параллельный умножитель  $4 \times 4$  требует для своей реализации 12 сумматоров. При дальнейшем наращивании разрядности матрица одноразрядных сумматоров значительно разрастается, одновременно увеличивается критический путь, и реализация умножителя становится нерациональной.

Одним из способов уменьшения аппаратных затрат служит использование алгоритма Бута. В соответствии с данным алгоритмом определённым образом анализируются парные биты множителя и в зависимости от их комбинации над множимым выполняются некоторые преобразования (пример для умножителя  $12 \times 12$  бит приведен в таблице 1).

**Таблица 1 – Преобразование множимого в зависимости от состояния множителя.**

| Пары бит множителя | Выполняемое действие             |
|--------------------|----------------------------------|
| 00                 | Прибавить 0                      |
| 01                 | Прибавить множимое               |
| 10                 | Прибавить ( $2 \times$ множимое) |
| 11                 | Прибавить ( $3 \times$ множимое) |

Сформированные частичные произведения, представляющие собой результат преобразования множимого, необходимо суммировать с соответствующим весовым фактором. Результатом их суммирования по алгоритму простого матричного умножителя и будет конечный продукт произведения двух операндов. Анализ различных методов свёртки частичных произведений показал, что наиболее приемлемым



для реализации на ПЛИС, является алгоритм на основе иерархического дерева многоразрядных масштабирующих сумматоров. При этом достигается сокращение аппаратных затрат по сравнению с параллельным матричным умножителем до двух раз.

Реализации данного решения для случая 12-разрядных операндов приведён на рис. 2. В данной схеме выбор соответствующего преобразования множимого  $Y$  осуществляется мультиплексором на основании анализа парных бит множителя  $X$  в соответствии с таблицей 1. Входные значения мультиплексоров  $0$  и  $Y$  подаются непосредственно, значение  $2Y$  формируется сдвигом множимого на один бит, а  $3Y$  формируется путём сложения  $Y$  и  $2Y$ .

Сумматоры ADD осуществляют суммирование частичных произведений, причём их разрядность для разных ступеней иерархии неодинакова. На каждом шаге суммирования, как видно, имеются несколько сквозных бит, непосредственно дополняющих выходные значения сумматоров. За счёт них и за счёт битов расширения знака осуществляется необходимое масштабирование результата. Размерность произведения  $P$  в конечном итоге равна сумме размерностей операндов.

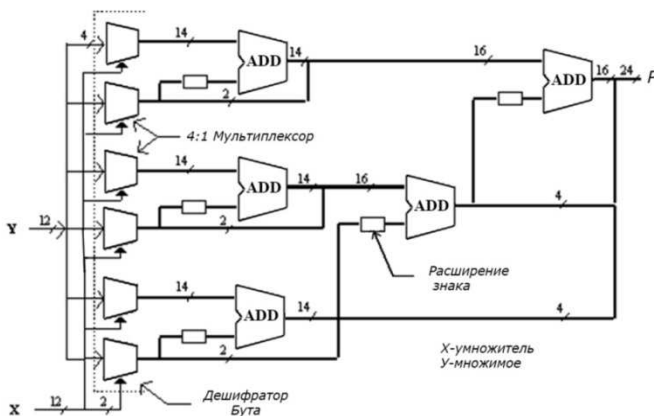


Рис. 2 – Структурная схема 12-разрядного умножителя на основе алгоритма Бута.

За счёт регулярности приведённой структуры быстродействие можно резко повысить путём введения конвейеризации каждой ступени. При этом разделяются конвейерными регистрами как сумматоры, так и мультиплексоры. Выигрыш в быстродействии благодаря данному решению оказывается особенно ощутимым для большой разрядности операндов.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Угрюмов Е.П. Цифровая схемотехника: Учеб. Пособие для вузов – СПб.:БХВ-Петербург, 2004. – 800 с.: ил.
2. Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов. 2-е изд. СПб.:Питер, 2011. – 688 с.: ил.

**Т.Ф. НГУЕН, А.П. ШИБАНОВ, Х.Л. ФАМ**

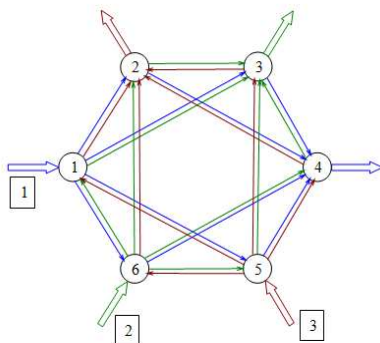
Вьетнамский государственный технический университет  
имени Ле Куй Дона

Рязанский государственный радиотехнический университет

### ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ ПЕРЕДАЧИ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ

*Рассматривается задача имитационного моделирования VLB-сети для передачи измерительной информации в программе Simulation.*

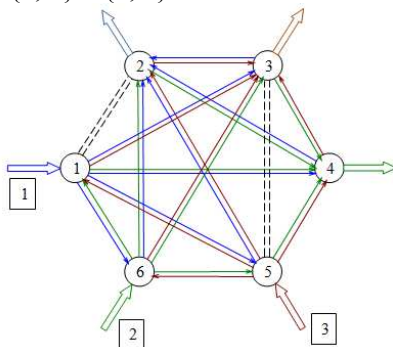
**Введение.** Сети с рандомизированным распределением нагрузки (VLB-сети) перспективны для создания инвариантных решений базовых опорных сетей [1]. Другими словами, такие опорные сети могут быть использованы без какой-либо модернизации различными поставщиками сетевых услуг (провайдерами). Узлы сети VLB образуют симметричную структуру с большим числом связей между ними. Пример такой сети приведен на рис. 1.



**Рис. 1 – Пример VLB-сети.**

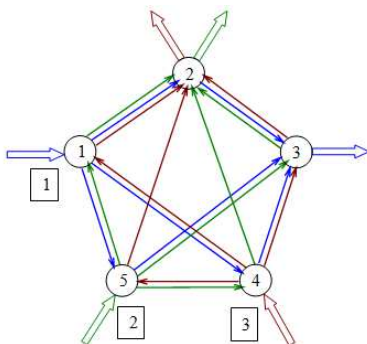
Симметричную нагрузку создают потоки 1, 2, 3, входящие в узлы 1, 6, 5. Передача каждого из этих потоков осуществляется по всевозможным двухфазным путям. Например, поток 1 передается в узел 4 одновременно по 4 путям: (1, 2, 3), (1, 3, 4), (1, 5, 4), (1, 6, 4). Увеличение числа путей для одновременной передачи кадра или потока позволяет повысить производительность сети, уменьшить время передачи и повысить ее надежность. При симметричной структуре узлов и одинаковых входных потоках загрузка линий связи VLB-сети будет одинаковой. Это идеальный случай. Если, например, на входы узлов VLB-сети будет подаваться потоки от измерительных средств, то будут созданы наилучшие из возможных условий для передачи информации. Чем больше число узлов VLB-сети, тем больше можно проложить двухфазных путей передачи кадров, содержащих измерительную информацию.

Идеальную симметрию структуры сети и поступающих потоков на практике получить проблематично. На рис. 2 изображена VLB-сеть с симметричной структурой и несимметричной нагрузкой во входных узлах. По сравнению с VLB-сетью, изображенной на рис. 1, произошло изменение нагрузки во внутренних линиях, в частности, оказались не нагруженными линии {1, 2} и {3, 5}.



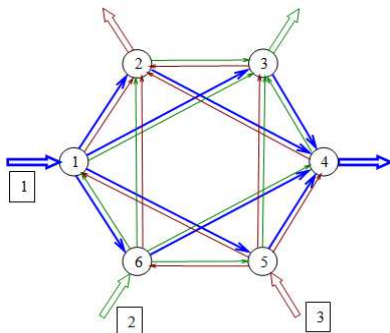
**Рис. 2 – VLB-сеть с симметричной структурой и разным направлением входных и выходных потоков.**

Для VLB-сети, составленной из 5 узлов, симметричное распределение внутренней нагрузки получить не удается (см. рис. 3).



**Рис. 3 – VLB-сеть с 5 узлами.**

Наличие неравномерной входной нагрузки VLB-сети (узел 1 и полосы пропускания в каналах для данного вида трафика показаны утолщенными линиями на рис. 4) даже в симметричной структуре приводит к неравномерной загрузке физических каналов.



**Рис. 4 – Неравномерная входная нагрузка в VLB-сети.**

Будем считать, что относительно неравномерное распределение нагрузки может иметь место на коротком отрезке времени. Если рассматривать систему в стационарном режиме, а изменения трафика оценивать как случайные, то на множестве реализаций матриц трафика и на бесконечно большом отрезке времени ( $t \rightarrow \infty$ ) можно считать, что потоки трафика (внешние и внутренние) усредняются.

Вместе с тем, для решения практических задач необходимо нужно определять требующиеся ресурсы полосы пропускания различных каналов. При изменении трафика в сети могут быть задействованы и зарезервированные каналы. Таким образом, в сети должна выполняться последовательность задач и загрузка каналов считаться усреднен-

ной на большом интервале времени. Для оценки требующейся полосы пропускания каналов и характеристик сети используем метод имитационного моделирования.

**Имитационное моделирование VLB-сети для передачи измерительной информации.** Для проведения моделирования используем программу Simulation разработки кафедры САПР РГРТУ. Ее использование обусловлено следующими причинами.

- доступностью исходного кода программы;
- тем фактом, что программа моделирования зарегистрирована в организации РОСПАТЕНТ Российской Федерации;
- быстротой составления моделей и удобством отладки, так как она имеет визуально-ориентированный интерфейс пользователя. Это означает, что облегчается поиск ошибок при отладке модели. Вместо того, чтобы искать ошибочные операторы в программе, достаточно визуально найти то место в изображении модели (дугу или имитационный блок), в котором процесс моделирования идет некорректно;
- система моделирования имеет интерфейс на уровне файловой системы с другими системами имитационного моделирования. Это позволяет создавать комбинированные системы моделирования, например, с системами GPSS, GERT Explorer; с сетями Петри;
- ориентацией на задачи моделирования телекоммуникаций.

Программа Simulation является разновидностью американской системы моделирования SLAM II (автор – Алан Прицкер). Сетевой подход (математическая сеть, состоящая из узлов и соединяющих их дуг) в исходной системе был реализован с использованием программных операторов. Более того, был даже создан графический интерфейс, но весьма примитивный – на основе операционной системы DOS. Программа Simulation в отличие от прототипа имеет современный графический интерфейс, развитые средства сбора статистики о проведенном эксперименте, множество функциональных блоков, ориентированных на моделирование телекоммуникаций, возможности построения комплексных систем моделирования.

Функциональные возможности имитационных блоков, в которые входят обрабатываемые заявки, будут раскрываться по мере необходимости при описании моделей каналов и сетей для передачи измерительной информации.

**Модель VLB-сети, состоящей из 5 узлов.** На рис. 5 изображена полная VLB-сеть, состоящая из 5 узлов, которые соединены между собой дуплексными каналами.

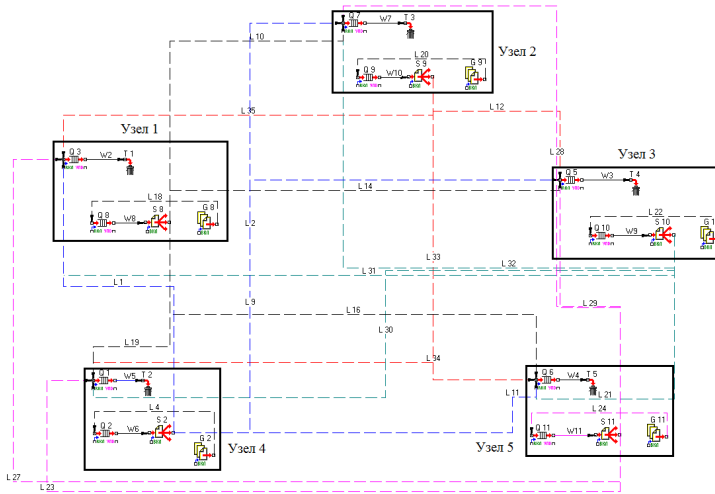


Рис. 5 – Полная VLB-сеть из 5 узлов.

На рис. 6 приведена конкретизация модели, в которой реализованы двухфазовые пути, показанные на рис. 7.

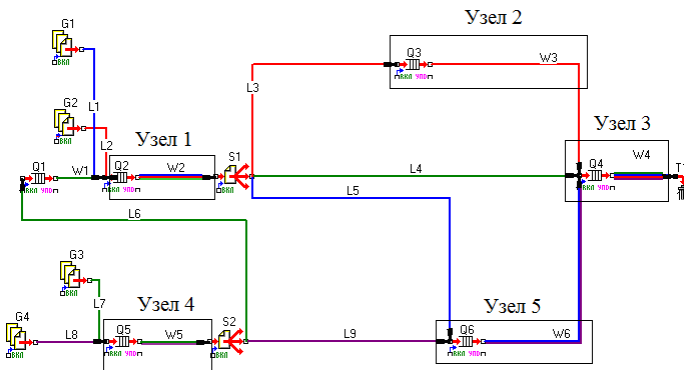
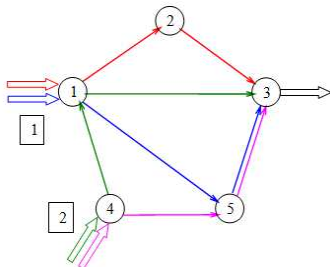


Рис. 6 – Поток в VLB-сети из 5 узлов.



**Рис. 7 – Направление потоков VLB-сети.**

Выходной поток узла 1 разделяется по двум путям: (1, 2, 3) и (1,5,3). Выходной поток узла 4 разделяется по путям (4, 1, 3) и (4, 5, 3).

Узел 1 на рис. 6 выполняет две функции: прием кадров от узла 4 и передачу собственных кадров. Имитацию входного буфера выполняет блок-очередь Q1, а задержек в коммутационном устройстве дуга W1. Выходной порт представлен парой Q2, W2. Это единственный узел, в котором имитируется входной поток (от узла 4) и выходной потоки в направлении узлов 2 и 5. Узел 4 представлен выходной функцией, узел 3 входной, а в узлах 2 и 5 определяются совокупные задержки и объем памяти в приемной и передающей частях канала и коммуникационного устройства. Заданы начальные значения: имитационные блоки-генераторы G1, G2, G3, G4 вырабатывают по 1000 кадров каждый с интервалом, определяемым экспоненциальным распределением с интенсивностью  $\lambda = 1$ . Начало генерации кадров соответственно в 0, 250, 500 и 750 ед. машинного времени. Задержки в дугах, помеченных буквой W, определяются экспоненциальной случайной величиной с интенсивностью  $\lambda = 1,5$ . По умолчанию блоки-очереди имеют размер памяти в 500 ед. Терминатор T1 определяет момент окончания моделирования после приема 4000 заявок.

Блоки-селекторы S1 и S2 реализуют ветвления в модели. Генераторы G1, G2, G3, G4 вырабатывают заявки, разделяющиеся по типам 0, 2, 1, 3 соответственно. Блоки-селекторы анализируют значения типов и направляют заявки с наименьшим значением типов на выходные дуги, имеющие наименьшие значения ASCII-кодов имен дуг. Селектор S1 направляет заявки типов 0 и 2 на дуги L3 и L5, а типа 1 на дугу L4. Селектор S2 направляет заявки 1 типа на дугу L6, а заявки типа 3 – на дугу L9.

Результаты моделирования приведены на рис. 8 – 11.

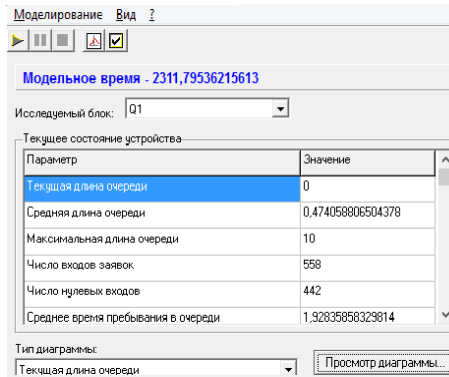


Рис. 8 – Экранная форма с табличными значениями.

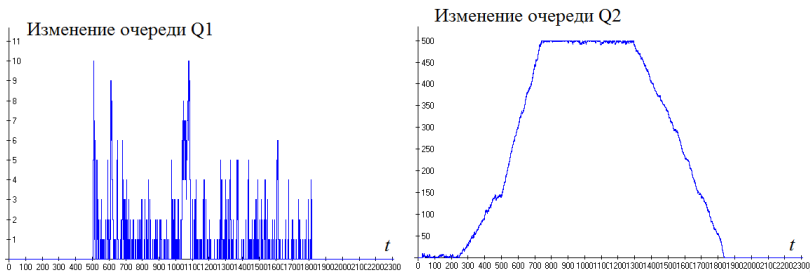


Рис. 9 – Диаграммы изменения состояния очередей Q1 и Q2.

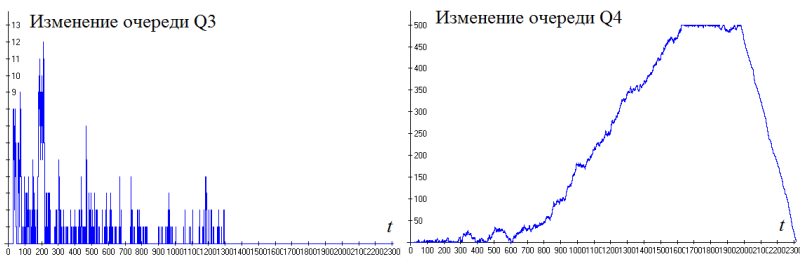
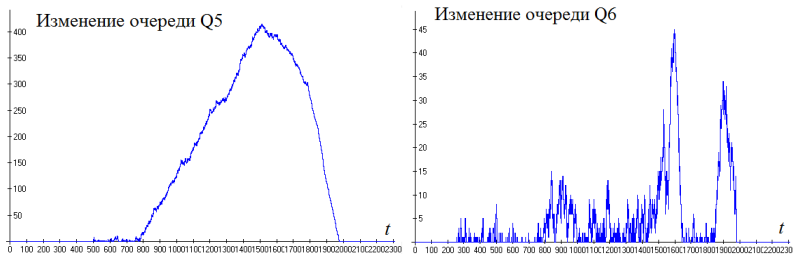


Рис. 10 – Диаграммы изменения состояния очередей Q3 и Q4.





**Рис. 11 – Диаграммы изменения состояния очередей Q5 и Q6.**

Очередь Q2 сильно перегружена, так как в этой точке сходятся потоки заявок, вырабатываемые всеми блоками-генераторами. Скачки трафика происходят в моменты времени 250, 500, 750 ед. Сильно перегружены и очереди Q4, Q5, Q6.

Как структура сети, так, и в особенности входные потоки не являются симметричными. Изначально полосы пропускания каналов недостаточны для передачи заданного трафика. Сеть передачи данных должна быть рассчитана на обработку любых возможных матриц трафика. В модели, приведенной на рис. 6, задана только одна матрица трафика.

Покажем, при каких параметрах каналов данная сеть успешно передаст заданный входной трафик и какой характер будет иметь изменение состояния очередей и их средние значения.

Значение интенсивности обработки во всех дугах увеличим от 1,5 до 5 ед. Это равносильно тому, что мы представим более производительные виртуальные каналы для потоков кадров в полосе пропускания физических каналов сети. После прогона модели с новыми параметрами получим средние значения величины очередей  $Q5 = 0,17$ ;  $Q1 = 0,03$ ;  $Q2 = 0,39$ ;  $Q4 = 0,82$ ;  $Q3 = 0,03$ ;  $Q6 = 0,11$ . Это обеспечивает успешную передачу кадров, и, как показывает практика, в том числе и в режиме реального времени. Диаграммы состояния очередей показаны на рис. 12 – 14. Диаграммы отражают влияние на величину очередей включение дополнительных потоков пакетов в моменты времени 0, 250, 500, 750, а также постепенное увеличение очереди Q4. Это естественно, так очередь Q4 находится на выходе системы, где сходятся все 4 потока.

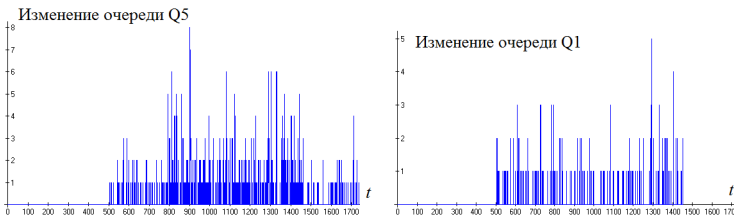


Рис. 12 – Диаграммы состояния очередей Q5 и Q1.

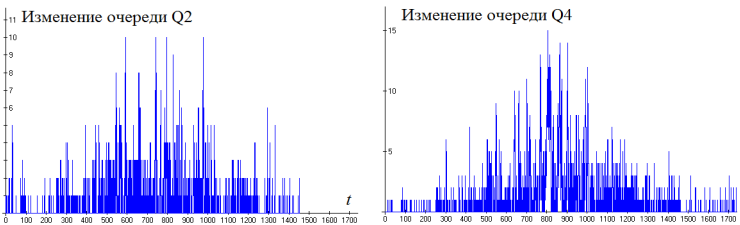


Рис. 13 – Диаграммы состояния очередей Q2 и Q4.

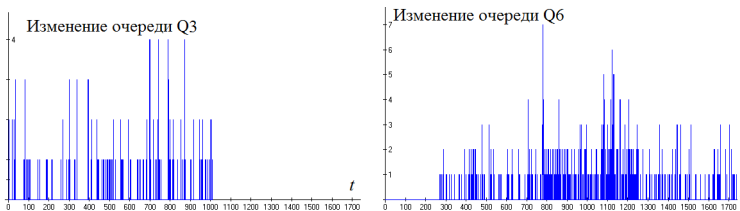


Рис. 14 – Диаграммы состояния очередей Q3 и Q6.

В заключение следует отметить, что данная методика позволяет определить нагрузку в каналах при любых входных матрицах трафика. Кроме того, она позволяет определить средние значения величин очередей в коммуникационных устройствах.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Ижванов Ю.Л., Корячко В.П., Шибанов А.П., Лукьянов О.В., Сапрыкин А.Н. Оптимизация сети с дозированной балансировкой нагрузки // Системы управления и информационные технологии. Москва-Воронеж, 2012. № 3(49). С. 37-42.

**И.Г. ОСИН, А.А. МАНАКИН**

Рязанский государственный радиотехнический университет

### **СРЕДСТВА РАСПОЗНАВАНИЯ РЕЧИ**

*Рассматриваются перспективные средства распознавания человеческой речи современными устройствами с целью практического применения.*

По мере развития компьютерных систем становится все более очевидным, что использование этих систем намного расширится, если станет возможным использование человеческой речи при работе непосредственно с компьютером, и в частности станет возможным управление машиной обычным голосом в реальном времени, а также ввод и вывод информации в виде обычной человеческой речи.

Существующие технологии распознавания речи не имеют пока достаточных возможностей для их широкого использования, но на данном этапе исследований проводится интенсивный поиск возможностей употребления коротких многозначных слов (процедур) для облегчения понимания. Распознавание речи в настоящее время нашло реальное применение в жизни, пожалуй, только в тех случаях, когда используемый словарь сокращен до 10 знаков, например, при обработке номеров кредитных карт и прочих кодов доступа в базирующихся на компьютерах системах, обрабатывающих передаваемые по телефону данные. Так что насущная задача - распознавание по крайней мере 20 тысяч слов естественного языка - остается пока недостижимой. Эти возможности пока недоступны для широкого коммерческого использования. Однако ряд компаний своими силами пытается использовать уже существующие в данной области науки знания.

Для использования функций речевого распознавания в различных устройствах, роботах, игрушках, разрабатываются аппаратные методы решения данной проблемы. Так американская компания Sensogy Inc. разработала интегральную схему Voice Direct™ 364 осуществляющую дикторозависимое распознавание небольшого числа команд (около 60) после предварительного обучения. Перед началом эксплуатации модуль необходимо обучить всем командам, используемым в работе. Команды сохраняются во внешнюю память в виде образов размером 128 байт. Во время работы, образ очередной команды сравнивается с эталонными из памяти в нейросетевом модуле и принимается решение о совпадении.

Тайваньская технологическая корпорация Primestar Technology Corporation разработала собственный чип VP-2025, предназначенный

для речевого распознавания. Данное устройство осуществляет распознавание с помощью нейросетевого метода.

Кроме того, американскими учеными принято решение создать специализированный микропроцессор для распознавания речи. Исследования в данном направлении будут проводиться сотрудниками Университета Карнеги-Меллон в Питсбурге (Пенсильвания) и Калифорнийского университета в Беркли. Ожидается, что новый микропроцессор появится в течение ближайших двух-трех лет. Причем эффективность распознавания речи таким чипом должна будет в 100-1000 раз превысить аналогичный показатель применяемых сегодня программно-аппаратных комплексов.

В App Store есть два бесплатных приложения для iPhone, iPad и iPod touch, позволяющие диктовать тексты SMS, писать электронной почтой, обновлений в соцсетях, а также задавать голосом поисковые запросы - Dragon Dictation и Dragon Search. Разработчик - американская компания Nuance, создатель решений для голосового управления компьютерами, а на русском языке - мультимедийными интерфейсами в автомобилях Mercedes, Audi, Toyota и Ford.

Dragon Dictation - приложение, работающее на основе технологии распознавания речи NaturallySpeaking. Оно ускоряет процесс набора текста до 5 раз! В нем имеется панель инструментов с соответствующими значками для быстрого доступа к электронной почте, SMS-сообщениям, статусам Facebook или Twitter, либо к буферу обмена. Приложение Dragon Dictation также включает в себя удобную функцию автоматического сохранения, которая не удаляет надиктованный текст, если пользователя во время диктовки прерывает входящий вызов. Благодаря мультязычности приложений пользователи могут без труда переключаться с одного языка на другой.

Dragon Dictation с поддержкой других языков уже загрузили несколько десятков миллионов пользователей по всему миру. Россия – 21-я страна, где компания Nuance локализует программы Dragon. И в каждой стране, где запускалось приложение, оно удерживалось на 1 или 2 месте рейтинга App Store 10 и даже попало в «Зал Славы» (Hall of Fame) App Store.

Dragon Search – это быстрый, простой и удобный способ поиска онлайн-контента для iPhone и iPad. Всё, что нужно, – ваш голос. Задайте вслух поисковый запрос, и приложение тут же выдаст результаты из основных ресурсов (Yandex, Google, iTunes, Wikipedia, YouTube). Специально разработанная визуальная оболочка «Dragon Search Carousel» позволяет переключаться между поисковыми

системами и с легкостью изменять текст запроса. Достаточно произнести: «Петр Ильич Чайковский» и нажать на вкладку iTunes – и тут же появятся результаты по этому запросу в онлайн-магазине. Перейдя на вкладки Яндекс, Wikipedia или Google можно ознакомиться с творческой биографией композитора

Одна из важнейших функций современного мобильного устройства - отправка текстовой информации. Для ввода оно в большинстве мобильных операционных систем используется экранная виртуальная клавиатура, которая в той или иной степени повторяет компьютерную. По сути, это прямой перенос технологий пока удовлетворяет большинство пользователей, однако немало компаний и ученых работает над тем, чтобы изобрести что-то дополняющее или полностью меняющее сам принцип ввода. Один из путей дополнения - голосовой ввод, о качественной поддержке которого заявляют все производители ОС.

Каждая из компаний-производителей популярных мобильных устройств использует разные технологии распознавания и превращения речи в текст. Google и Microsoft разрабатывают собственные, тогда как Apple развивает то, на что приобрела лицензию у Nuance, знакомой владельцам Android-устройств по технологии Dragon Dictation, встроенной в некоторые сторонние проекты.

Android уже давно содержит функцию оффлайнового голосового ввода, и по всей видимости, Apple стремится реализовать эту функцию, чтобы повысить свою конкурентоспособность по отношению к ОС Google. Аналогичная функция существует в OS X Mavericks в виде дополнительно устанавливаемого пакета. Вполне возможно, что будущая поддержка оффлайнового голосового ввода в iOS может потребовать также дополнительной загрузки и установки программных компонентов.

Как известно, Apple давно сотрудничает с Nuance Communication – одним из ключевых игроков на рынке решений для распознавания голоса и управления устройствами при помощи речи. Но по сей день мобильная система iOS не предлагает каких-либо расширенных возможностей по голосовому взаимодействию с пользователем.

Было бы неправильно считать, что именно Apple «сделает сказку былью», так как, планшетные компьютеры были и задолго до выхода iPad, а попытки научить электронные устройства понимать голос человека предпринимаются с самого появления этих устройств. Однако, именно технологии и разработки компании из Cupertino находят массовое применение.

Как видно, технологии речевого распознавания нашли свое применение в различных областях. Однако в данной области множество проблем все еще остаются не решенными, многие идеи требуют дальнейшего развития. Так, программы, работающие с изолированными словами, достигли высокой точности в командных системах – в наиболее распространенных современных приложениях точность распознавания составляет в среднем 95-99% и зависит в основном от уровня шума. В то же время задача распознавания слитной речи в достаточной степени не решена, хотя в случае ограниченного словаря системы такого типа существуют (VoxReports на ядре ViaVoice, Verbmobil) и показывают высокие результаты по точности. В настоящее время множество работ посвящено проблеме распознавания слитной речи, так как именно такой тип речевого взаимодействия считается наиболее перспективным.

#### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. <https://lifehacker.ru/2011/09/02/dragon-dictation-dlya-ios-prevrashhaet-ustnuyu-rech-v-pechatnoe-slovo/>
2. <http://documentslide.com/documents/vp-2025-developing-tool-kit-55954b941649d.html>

**Д.А. ПЕРЕПЕЛКИН, В.С. БЫШОВ**

Рязанский государственный радиотехнический университет

#### **АДАПТИВНАЯ МАРШРУТИЗАЦИЯ В ПРОГРАММНО- КОНФИГУРИРУЕМЫХ СЕТЯХ**

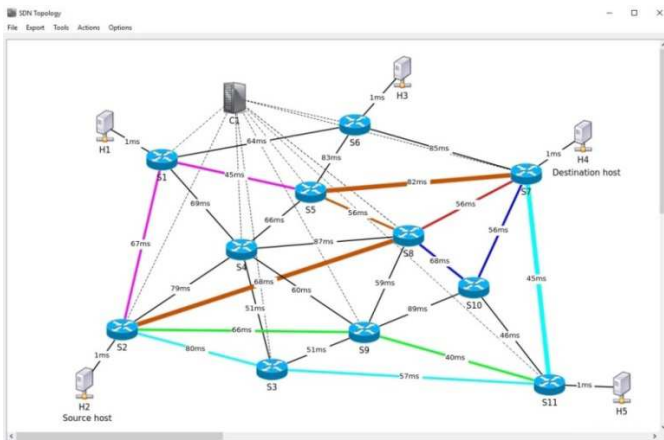
*Разработана система моделирования процессов адаптивной маршрутизации в программно-конфигурируемых сетях и показана ее работа на примере алгоритма многопутевой адаптивной маршрутизации – алгоритма Йена.*

В настоящее время увеличивается количество сетевых устройств, реализующих передачу различных видов данных. Все это меняет вид компьютерных сетей (КС) и ведет к усложнению функций управления потоками данных и не позволяет обеспечить требуемое качество сетевых сервисов (Quality of Service, QoS). Ответом на кризис КС стало появление принципиально нового подхода к их построению – программно-конфигурируемые сети (ПКС) [1]. Архитектуру ПКС можно представить, как множество простых программ и аппаратных коммутаторов, подключенных к общему контроллеру (Controller

OpenFlow), реализованных в виде обычного сервера со специальным программным обеспечением.

ПКС позволяет успешно реализовать и интегрировать различные алгоритмы адаптивной маршрутизации, широко используемые в современных КС [2]. Такие алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Среди алгоритмов адаптивной маршрутизации на практике чаще всего используют алгоритм Дейкстры. Однако для повышения качества сетевых сервисов и увеличения отказоустойчивости КС в последнее время используют многопутевую адаптивную маршрутизацию. Протоколы многопутевой маршрутизации, как правило, используют в своей работе алгоритм  $k$ -кратчайших путей (алгоритм Йена). Многопутевая маршрутизация позволяет сбалансировать нагрузку на сеть путем распределения сетевого трафика по резервным маршрутам.

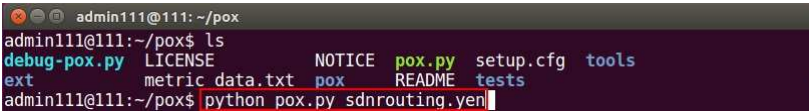
Для исследования алгоритмов адаптивной маршрутизации в ПКС на основе протокола OpenFlow, который управляет потоками данных между сетевыми устройствами и контроллером в ПКС, разработана специализированная визуальная среда SDN Topology. Программное средство позволяет проектировать и исследовать различные топологии в ПКС в режиме реального времени с возможностью интеграции данных в Mininet. Интерфейс программной среды и результаты работы алгоритма Йена представлены на рис. 1.



**Рис. 1 – Программный интерфейс и результат работы алгоритма Йена.**

Алгоритм Йена, в отличие от алгоритма Дейкстры, находит не один, а  $k$ -кратчайших маршрутов между двумя вершинами графа. Дан-

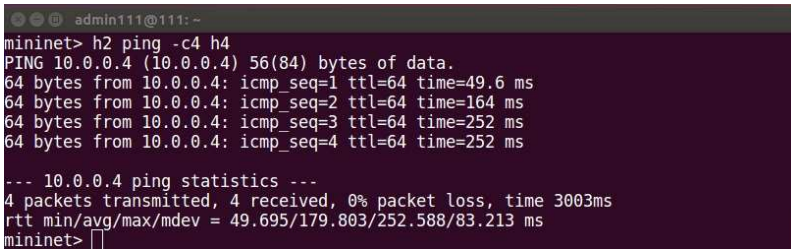
ный параметр задает максимальное количество кратчайших путей, вычисляемых по алгоритму Йена. Каждый отдельный кратчайший путь в соответствии с алгоритмом Йена находится по алгоритму Дейкстры. Для того, чтобы ПКС-контроллер осуществлял маршрутизацию, необходимо запустить модуль алгоритма Йена при запуске контроллера (Рис. 2).



```
admin111@111: ~/pox
admin111@111:~/pox$ ls
debug-pox.py  LICENSE          NOTICE  pox.py  setup.cfg  tools
ext           metric data.txt  pox      README  tests
admin111@111:~/pox$ python pox.py sdnrouting.yen
```

**Рис. 2 – Запуск контроллера совместно с модулем алгоритма Йена.**

Далее следует запустить проверку соединения хоста Н2 с хостом Н4, как показано на рис. 3.



```
mininet> h2 ping -c4 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=49.6 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=164 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=252 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=252 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 49.695/179.803/252.588/83.213 ms
mininet>
```

**Рис. 3 – Проверка соединения между Н2 и Н4 для запуска алгоритма Йена.**

Следует отметить, что пакеты передаются по самому кратчайшему маршруту, который в данном случае совпадает маршрутом из примера с алгоритмом Дейкстры. Остальные найденные маршруты могут быть использованы, например, для балансировки нагрузки. Установка маршрутов в ПКС, найденных с помощью алгоритма Йена, изображено на рис. 4.



```

admin111@111: ~/pox
sdnrouting_yen_routing Received packet [SRC: h2(00:00:00:00:00:02) DST: h4(00:00:00:00:00:04)]
sdnrouting_yen_routing Received packet [SRC: h4(00:00:00:00:00:04) DST: h2(00:00:00:00:00:02)]
sdnrouting_yen_routing Computed route from h4 to h2: s7->s8->s2
sdnrouting_yen_routing Computed route from h4 to h2: s7->s11->s9->s2
sdnrouting_yen_routing Computed route from h4 to h2: s7->s8->s9->s2
sdnrouting_yen_routing Computed route from h4 to h2: s7->s11->s3->s2
sdnrouting_yen_routing Computed route from h4 to h2: s7->s5->s1->s2
sdnrouting_yen_routing Computed route from h4 to h2: s7->s6->s1->s2
sdnrouting_yen_routing Installing forward from switch s7 to output port 6
sdnrouting_yen_routing Installing forward from switch s7 to switch s8 output port 3
sdnrouting_yen_routing Installing forward from switch s8 to switch s2 output port 1
sdnrouting_yen_routing Received packet [SRC: h2(00:00:00:00:00:02) DST: h4(00:00:00:00:00:04)]
sdnrouting_yen_routing Computed route from h2 to h4: s2->s8->s7
sdnrouting_yen_routing Computed route from h2 to h4: s2->s9->s11->s7
sdnrouting_yen_routing Computed route from h2 to h4: s2->s3->s11->s7
sdnrouting_yen_routing Computed route from h2 to h4: s2->s8->s10->s7
sdnrouting_yen_routing Computed route from h2 to h4: s2->s1->s5->s7
sdnrouting_yen_routing Computed route from h2 to h4: s2->s8->s5->s7
sdnrouting_yen_routing Installing forward from switch s2 to output port 6
sdnrouting_yen_routing Installing forward from switch s2 to switch s8 output port 4
sdnrouting_yen_routing Installing forward from switch s8 to switch s7 output port 3

```

Рис. 4 – Установка маршрутов в ПКС.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. Openflow: Enabling Innovation in Campus Networks Proc. ACM SIGCOMM Computer Communication Review. 2008. Vol. 38. No. 2, pp. 69-74.
2. Перепелкин Д.А., Перепелкин А.И. Разработка алгоритмов адаптивной маршрутизации в корпоративных вычислительных сетях // Вестник Рязанского государственного радиотехнического университета. 2006. №19. С. 114-116.

Д.А. ПЕРЕПЕЛКИН, И.Ю. ЦЫГАНОВ

Рязанский государственный радиотехнический университет

### КОМБИНИРОВАННЫЙ АЛГОРИТМ ДЕЙКСТРЫ И СЕГМЕНТАЦИИ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

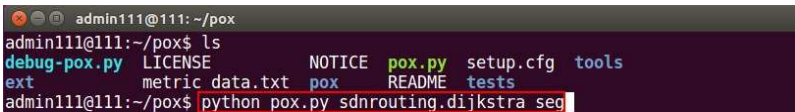
*В работе проведено исследование работы алгоритма Дейкстры и алгоритма сегментации в программно-конфигурируемых сетях на основе протокола OpenFlow.*

Стремительное развитие вычислительной производительности, объемов памяти и возможностей высокоскоростной передачи данных привело к необходимости разработки новых технологий построения сетевых инфраструктур. Программно-конфигурируемые сети (ПКС) – развивающаяся архитектура сети, где функция управления сетью разделена с функцией передачи данных и полностью программируема [1].

Нами разработана визуальная среда проектирования SDN Topology, предназначенная для создания и редактирования топологий ПКС и последующего запуска в эмуляторе MiniNet [2]. Среда доступна на платформах Linux и Windows.

Для исследования алгоритмов маршрутизации в ПКС мы используем контроллер POX. По своей сути POX – это платформа для быстрой разработки и прототипирования ПО управления сетью. Контроллер POX написан на языке Python [3].

Разработанный нами алгоритм сегментации [4] позволяет разделить программно-конфигурируемую сеть на виртуальные области. Наличие таких областей может значительно уменьшить трудоемкость расчета кратчайшего пути. Рассмотрим работу алгоритма сегментации в комбинации с работой алгоритма Дейкстры [5]. Пусть нужно найти кратчайший маршрут от хоста H2 до хоста H4 (рисунок 3). Запуск контроллера с подключенным модулем алгоритма сегментации и алгоритма Дейкстры приведен на рис. 1.



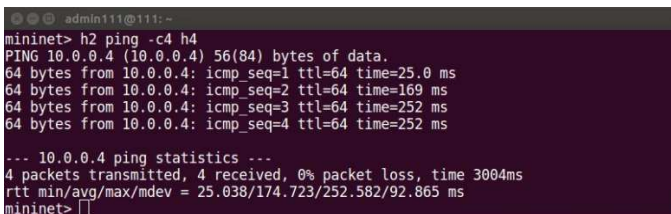
```

admin111@111:~/pox
admin111@111:~/pox$ ls
debug-pox.py  LICENSE          NOTICE  pox.py  setup.cfg  tools
ext           metric.data.txt  pox      README  tests
admin111@111:~/pox$ python pox.py sdnrouting.dijkstra seg
  
```

**Рис. 1 – Запуск ПКС-контроллера POX с подключенным модулем алгоритма Дейкстры и алгоритма сегментации.**

Эмулятор MiniNet запускается так же, как и в предыдущем примере. Попытка проверить соединения между хостом H2 и хостом H4 успешна, как показано на рис. 2.

Поясним работу комбинированного алгоритма Дейкстры и сегментации. Первый посланный пакет поступает в коммутатор S2. Если в коммутаторе S2 не установлены правила обработки полученного пакета, то копия данного пакета отправляется контроллеру для анализа и получения необходимых инструкций. Контроллер, в свою очередь, запускает следующий алгоритм.



```

mininet> h2 ping -c4 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=25.0 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=169 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=252 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=252 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 25.038/174.723/252.582/92.865 ms
mininet>
  
```

**Рис. 2 – Проверка соединения между H2 и H4 для запуска алгоритма Дейкстры и алгоритма сегментации.**

**Шаг 1.** Проверить, имеются ли данные о виртуальных областях. Если множество виртуальных областей пусто, то запустить алгоритм сегментации, который сформирует непустое множество виртуальных областей. Иначе перейти к шагу 2.

**Шаг 2.** Проанализировать IP адрес отправителя и получателя. Если и отправитель, и получатель в одной виртуальной области, то запустить алгоритм Дейкстры, передав в качестве множества узлов множество узлов данной виртуальной области. Иначе запустить алгоритм Дейкстры, передав в качестве множества узлов множество всех узлов ПКС.

Таким образом, если отправитель и получатель пакета находятся в одной виртуальной области, то трудоемкость расчета кратчайшего пути по алгоритму Дейкстры уменьшается, так как мощность множества узлов отдельной виртуальной области всегда меньше, чем мощность множества узлов всей ПКС.

Результат работы комбинированного алгоритма Дейкстры и сегментации можно увидеть на рис. 3. Как видно из рисунка, сеть разделилась на две виртуальной области. Поскольку коммутаторы S2 и S7 расположены в одной такой области, расчет кратчайшего пути производился только для графа этой области, что уменьшило общую вычислительную сложность поиска пути от H2 до H4.

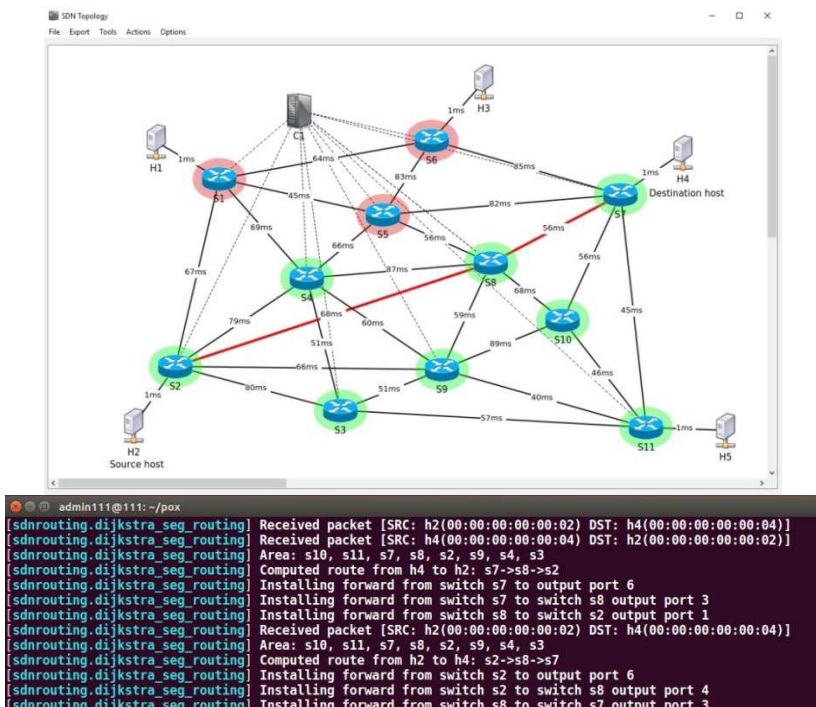


Рис. 3 – Результат работы комбинированного алгоритма Дейкстры и сегментации.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Software-defined networking: the new norm of Networks, ONF White Paper. April 2012.
2. Эмулятор компьютерных сетей MiniNet [Электронный ресурс] <http://mininet.org>.
3. Язык программирования Python [Электронный ресурс] <https://www.python.org>.
4. Перепелкин Д.А., Цыганов И.Ю.. Усовершенствованный алгоритм сегментации структур корпоративных сетей по критерию минимальной стоимости // Вестник Рязанского государственного радиотехнического университета. 2015. № 53. С. 48-57.
5. Dijkstra E.W.. A note on two problems in connexion with graphs. Numer. Math – Springer Science + Business Media, 1959. Vol 1, iss, 1, pp. 269-271.

**А.В. ПЕРОВ**

Рязанский государственный радиотехнический университет

## **ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ ПРИ РЕАЛИЗАЦИИ АЛГОРИТМОВ**

*Рассматривается одно из перспективных средств увеличения производительности нахождения решения алгоритмов – применение параллельного программирования.*

При разработке алгоритмов очень часто встает вопрос об адекватности времени нахождения решения для конкретных исходных данных. Так, например, классическая задача коммивояжера, в зависимости от количества вершин графа городов, может решаться как за секунды, так и за тысячелетия. При больших размерностях задач принято либо упрощать задачи, либо находить новые алгоритмы и способы нахождения решения, что порой бывает не только сложно, но и попросту невозможно. Но что делать, если задача решается слишком быстро чтобы придумывать новый алгоритм, но и слишком долго чтобы оставлять алгоритм без изменений? В данном случае огромную пользу может принести разработка алгоритма, адаптированного под параллельное выполнение.

Данный подход в программировании применяется для увеличения степени параллелизма в программе, что позволяет выполнять отдельные участки кода одновременно. Данный подход порой значительно усложняет реализацию, однако за счет него возможно существенное увеличение производительности алгоритма.

Существует несколько способов реализации параллельных вычислений:

- 1) Реализация в виде процесса операционной системы;
- 2) Реализация в виде набора потоков одного процесса.

При использовании реализации в виде процесса операционной системы программист обладает большими возможностями, однако данная реализация отличается большими накладными расходами при переключении контекста процесса.

В случае реализации в виде набора потоков программист не имеет доступа к некоторым полезным средствам операционной системы, но он получает возможность создания большого количества параллельно-выполняемых участков кода, доступ к общей для всех потоков разделяемой памяти процесса, а также удобному программному интерфейсу, реализованному практически во всех языках программирования.

Для повышения степени параллелизма в алгоритме, в первую очередь, необходимо выделить блок действий, который периодически повторяется, и оформить его в качестве функции или процедуры. От правильности выделения этой функции зависит производительность программы, так как эта функция в дальнейшем будет выполняться в отдельном потоке на отдельном ядре процессора. Таким образом, применимость параллельного программирования в реализации алгоритмов во многом обусловлена повторяемостью действий в самом алгоритме.

Рассмотрим несколько алгоритмов и применимость параллельного программирования в них.

Одним из основных алгоритмов дискретной оптимизации является метод ветвей и границ. В основе этого метода лежит нахождение верхней или нижней граничной оценки решения – корня дерева, с последующим ветвлением и образованием границы – списка веток или листьев дерева ветвления. В качестве повторяемого действия в данном алгоритме отлично подходит процесс ветвления дерева, который можно выполнять одновременно в разных потоках. Однако, в методе ветвей и границ мы всегда имеем дело с постоянным изменением границы, мы добавляем и удаляем вершины, что ведет к серьезным ограничениям параллельного выполнения. Как отмечалось выше, параллельное программирование может значительно усложнить реализацию алгоритма, и в данном случае необходимо корректно реализовать доступ к границе из разных потоков. Каждое подобное усложнение реализации алгоритма ведет к неизбежному снижению скорости выполнения и увеличению потребляемой памяти, что является полной противоположностью желаемого результата. Именно поэтому необходимо здраво оценить соотношение дополнительных накладных расходов к доступным вычислительным мощностям и принять решение о необходимости повышения параллелизма в реализации алгоритма.

Хорошим примером алгоритма, подходящим для параллельного выполнения, является алгоритм, реализующий идеи метода динамического программирования. Ключевой идеей метода является предположение, что для решения задачи требуется решить отдельные ее части и после этого объединить их в одно решение. Так как эти подзадачи в большинстве своем одинаковы, можно составить рекуррентное соотношение, код которого можно выполнять в отдельных потоках. Однако, как и в методе ветвей и границ, встает вопрос о доступе к разделяемому ресурсу – графу решений подзадач, что требует синхронизации доступа и, соответственно, усложнения реализации алгоритма. Тем не менее, можно заметить, что данный граф изменяется только

при добавлении новых вершин, что в значительной степени упрощает процесс синхронизации конкурентного доступа к нему.

Необходимо отметить, что реализация алгоритма с высокой степенью параллелизма не гарантирует увеличения производительности, однако она значительно увеличивает возможности горизонтального масштабирования алгоритма для выполнения на многоядерных процессорах, многопроцессорных системах, GPU и даже в промышленных вычислительных центрах.

Таким образом, можно выделить несколько правил применения параллельного программирования в реализации алгоритмов:

- 1) необходимо правильно выбрать функцию, которая будет выполняться в отдельных потоках;
- 2) необходимо оценить вычислительные мощности оборудования, на котором будет выполняться код алгоритма – добавление параллелизма в алгоритм далеко не всегда принесет повышение его производительности, а в некоторых случаях даже уменьшит ее;
- 3) необходимо понимать принципы многопоточного программирования для правильной оценки требуемых процессорного времени и оперативной памяти.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: учебное пособие. – Изд. 2-е, испр. – М.: ФИЗМАТЛИТ, 2003. С. 95 - 137
2. Goetz B., Peierls T., Bloch J., Bowbeer J., Holmes D., Lea D. Java Concurrency in Practice. 1st Edition - Addison-Wesley Professional, 2006

### **П.С. ПИСАКА**

Рязанский государственный радиотехнический университет

### **АЛГОРИТМ ПРИВЯЗКИ ФАЙЛОВ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ НА ОСНОВЕ ВЗАИМНОГО АНАЛИЗА БОРТОВОГО И СТАНЦИОННОГО ВРЕМЕНИ**

*Предложен многокритериальный алгоритм для решения задачи взаимной привязки файлов телеметрической информации (ТМИ) процедуры монтажа единого носителя ТМИ в целях повышения качества.*

В процессе испытаний изделий ракетно-космической техники на активном участке полета осуществляется передача телеметрической информации (ТМИ) о состоянии размещенных на борту систем. Для

приема и регистрации ТМИ, на протяжении всей траектории расположены малогабаритные приемно-регистрирующие станции. Для объективной оценки состояния работы изделия в процессе полета, а также по его результатам необходимо проведение качественного анализа принятой ТМИ. Следовательно, особую роль в системах автоматизации испытаний изделий ракетно-космической техники играет процедура монтажа единого носителя ТМИ [1]. Цель процедуры заключается в повышении качества ТМИ, зарегистрированной на различных приемных станциях.

Решение задачи взаимной привязки файлов ТМИ – один из важных этапов указанной процедуры [2]. Структурной единицей ТМИ является кадр, а кадры имеют одинаковую структуру [3]. Кадр состоит из информационных каналов, содержащих измерительную информацию с датчиков, установленных на борту изделия. При этом в служебных каналах передаются маркеры, по значениям которых можно судить о правильности структуры принятой ТМИ. Цель взаимной привязки файлов ТМИ заключается в определении блоков ТМИ из разных файлов, соответствующих различным приемным станциям и содержащих одну и ту же переданную через эфир информацию.

Цель работы – разработка многокритериального алгоритма взаимной привязки файлов ТМИ от территориально распределенных приемных станций.

Пусть  $\mathbf{F} = \{f_i\}$ ,  $i = 1, \dots, N$  множество зарегистрированных файлов ТМИ от территориально распределенных приемных станций, где  $N$  количество файлов ТМИ, а  $f_i$  - индекс зарегистрированного файла ТМИ. Алгоритм взаимной привязки файлов ТМИ состоит из двух этапов:

1. Разбиение файлов на непересекающиеся множества;
2. Определение индексов блоков информации из разных файлов, содержащих одну и ту же информацию, передаваемую через эфир.

На первом этапе алгоритма взаимной привязки осуществляется разбиение множества файлов  $\mathbf{F}$  на непересекающиеся подмножества файлов  $\mathbf{F}_j = \{f_{j,k}\}$ ,  $j = 1, \dots, J$ ,  $k = 1, \dots, K_j$ , где  $J$  - количество подмножеств, а  $K_j$  - количество файлов в подмножестве  $\mathbf{F}_j$ , причем

$$\sum_{j=1}^J K_j = N. \text{ Таким образом, выполняется условие}$$



$$\left\{ \begin{array}{l} \mathbf{F} = \bigcup \mathbf{F}_j, \quad j = 1, \dots, J, \\ \mathbf{F}_{j_1} \cap \mathbf{F}_{j_2} = 0, \quad j_1 = 1, \dots, J, \quad j_2 = 1, \dots, J, \quad j_1 \neq j_2. \end{array} \right. \quad (1)$$

Каждый файл  $f_{j,k}$  внутри подмножества  $\mathbf{F}_j$ ,  $j = 1, \dots, J$  связан непосредственно или опосредованно через другие файлы с любым другим файлом из этого же подмножества. Файлы из разных подмножеств не могут быть связаны непосредственно или опосредовано. Непосредственно связанными считаются те файлы, которые содержат одну и ту же информацию, передаваемую через эфир с высокой степенью вероятности.

Рассмотрим подмножество файлов ТМИ  $\mathbf{F}_j$ ,  $j = 1, \dots, J$ . Каждый файл ТМИ  $f_k$  подмножества  $\mathbf{F}_j$  может быть представлен как множество кадров  $C_w^k$ , т.е.  $\mathbf{C}_k = \{C_w^k\}$ ,  $w = 1, \dots, W_k$ , где  $W_k$  - количество кадров в  $k$ -ом файле. Каждый кадр  $C_w^k$ , характеризуется значениями станционного времени  $s_w^k$ , бортового времени  $b_w^k$  и множеством значений в каналах кадра  $\mathbf{R}_w^k$ . Множество значений в каналах кадра  $\mathbf{R}_w^k$  может быть представлено в виде  $\mathbf{R}_w^k = \{r_u^{k,w}\}$ ,  $u = 1, \dots, U$ , где  $r_u^{k,w}$  - значение в  $u$ -ом канале кадра  $C_w^k$ ,  $U$  - количество каналов в кадре  $C_w^k$ .

Для анализа возможности привязки по времени файла  $f_k$  к файлу  $f_n$ , где  $f_k \in \mathbf{F}_j$ ,  $f_n \in \mathbf{F}_j$ ,  $k = 1, \dots, K_j$ ,  $n = 1, \dots, K_j$ ,  $n \neq k$ , используются фрагменты (последовательности кадров) ТМИ, для которых должны выполняться следующие условия:

1. Из файлов  $f_k$  и  $f_n$  можно выделить подмножества бортовых времен  $\{b_{x_1}^k\}$ ,  $x_1 = 1, \dots, X_1$  и  $\{b_{x_2}^k\}$ ,  $x_2 = 1, \dots, X_2$ , где  $X_1$  и  $X_2$  - количество кадров файлов, для которых бортовое время  $b_{x_1}^k$  и  $b_{x_2}^n$  корректно (с каждой секундой возрастает на единицу или обнуляется), должно быть не меньше заданного значения  $X$ . Таким образом, должно выполняться условие первого критерия

$$\left\{ \begin{array}{l} X_1 \geq X, \exists \{b_{x_1}^k\}, \forall b_{x_1}^k \left[ \begin{array}{l} b_{x_1+1}^k - b_{x_1}^k = 1 \\ b_{x_1}^k = 0 \end{array} \right], x_1 = 1, \dots, X_1 - 1, \\ X_2 \geq X, \exists \{b_{x_2}^n\}, \forall b_{x_2}^n \left[ \begin{array}{l} b_{x_2+1}^n - b_{x_2}^n = 1 \\ b_{x_2}^n = 0 \end{array} \right], x_2 = 1, \dots, X_2 - 1. \end{array} \right. \quad (2)$$

2. Из файлов  $f_k$  и  $f_n$  можно выделить подмножества стационарных времен  $\{s_{y_1}^k\}$ ,  $y_1 = 1, \dots, Y_1$  и  $\{s_{y_2}^n\}$ ,  $y_2 = 1, \dots, Y_2$ , где  $Y_1$  и  $Y_2$  - количество кадров файлов, для которых стационарное время  $s_{y_1}^k$  и  $s_{y_2}^n$  корректно (разность стационарных времен соседних кадров должна принадлежать заданному допусковому интервалу  $l = [l_1; l_2]$ ), должно быть не меньше заданного значения  $Y$ . Таким образом, должно выполняться условие второго критерия

$$\left\{ \begin{array}{l} Y_1 \geq Y, \exists \{s_{y_1}^k\}, \forall s_{y_1}^k (s_{y_1+1}^k - s_{y_1}^k) \in l, y_1 = 1, \dots, Y_1 - 1, \\ Y_2 \geq Y, \exists \{s_{y_2}^n\}, \forall s_{y_2}^n (s_{y_2+1}^n - s_{y_2}^n) \in l, y_2 = 1, \dots, Y_2 - 1. \end{array} \right. \quad (3)$$

3. Качество кадров  $Q_k$  и  $Q_n$  файлов  $f_k$  и  $f_n$  должно быть не меньше заданного значения  $Q$ . Таким образом, должно выполняться условие третьего критерия

$$\left\{ \begin{array}{l} Q_k \geq Q, \\ Q_n \geq Q. \end{array} \right. \quad (4)$$

В качестве критерия качества могут быть использованы: бит четности, бит чередования, контрольная сумма.

Два кадра  $C_{x_1}^k$  и  $C_{x_2}^n$  из разных файлов  $f_k$  и  $f_n$  считаются идентичными, т.е. содержащими одну и ту же переданную через эфир информацию, если выполняются условия:

1. Кадры  $C_{x_1}^k$  и  $C_{x_2}^n$  принадлежат фрагментам ТМИ, для которых выполняются критерии (2), (3), (4);

2. Для кадров  $C_{x_1}^k$  и  $C_{x_2}^n$  совпадают значения бортового времени, т.е.  $b_{x_1}^k = b_{x_2}^n$ ;

3. Для кадров  $C_{x_1}^k$  и  $C_{x_2}^n$  доля совпавших информационных бит, т.е. бит, передаваемых через эфир, и не сформированных приемной станцией, выше заданной величины  $Z$ .

Считается, что два файла  $f_k$  и  $f_n$  могут быть привязаны, если они содержат с высокой степенью вероятности одну и ту же передаваемую через эфир информацию. При этом кадры из файлов с одной и той же информацией могут быть идентифицированы, если выполняются условия:

1. Из подмножеств  $\{C_{x_1}^k\}$  и  $\{C_{x_2}^n\}$ , количество пар кадров  $C_{x_1}^k$  и  $C_{x_2}^n$  для которых выполняется условие идентичности, не ниже заданного значения  $A$ .

2. Максимальная погрешность восстановления станционного времени  $D$ , обусловленная разной скоростью хода часов на разных станциях регистрации, должна быть меньше заданного значения.

Критерий максимальной погрешности  $D$  определяется путем анализа станционных времен для идентичных кадров ТМИ. Пусть имеется множество  $S$  из  $G$  пар станционных времен  $(s_{y_1}^k, s_{y_2}^n)$  для идентичных кадров подмножеств  $\{C_{y_1}^k\}$  и  $\{C_{y_2}^n\}$  файлов  $f_k$  и  $f_n$ . Это множество можно представить выражением

$$S = \{(s_{y_1}^k, s_{y_2}^n)\}; \quad y_1 = 1, \dots, G, \quad y_2 = 1, \dots, G. \quad (5)$$

Можно ввести множество  $D$  разностей значений станционных времен идентичных кадров, представленное выражением

$$D = \{d_g\}; \quad d_g = s_{y_1}^k - s_{y_2}^n, \quad y_1 = 1, \dots, G, \quad y_2 = 1, \dots, G. \quad (6)$$

Отсюда, исходя из выражений вида (5) и (6), критерий максимальной погрешности  $D$  можно представить выражением

$$D = \max(|d_{g+1} - d_g|); \quad g = 1, \dots, G-1. \quad (7)$$

Экспериментальная отработка предложенного алгоритма проводилась при обработке ТМИ пяти приемных станций по результатам пусков изделий ракетно-космической техники.

Таким образом, предложенный алгоритм позволяет, с использованием нескольких критериев, осуществлять взаимную привязку файлов ТМИ зарегистрированных на различных приемных станциях. Полученные в результате работы алгоритма файлы используются на дальнейших этапах процедуры монтажа единого носителя ТМИ [1].

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Зорин С.В., Писака П.С. Процедура монтажа единого носителя телеметрической информации // Материалы XIX Всероссийской научно-технической конференции студентов, молодых ученых и специалистов «Новые информационные технологии в научных исследованиях и в образовании». Рязань: Рязанский государственный радиотехнический университет, 2014. С. 109-111.

2. Зорин С.В., Писака П.С. Задача взаимной привязки файлов телеметрической информации // Материалы XX Юбилейной Всероссийской научно-технической конференции студентов, молодых ученых и специалистов «Новые информационные технологии в научных исследованиях». Рязань: Рязанский государственный радиотехнический университет, 2014. С. 107-108.

3. Кириллов С.Н., Зорин С.В., Писака П.С. Алгоритм выделения кадров из потока зарегистрированной телеметрической информации // Вестник Рязанского государственного радиотехнического университета. 2015. № 54 часть 2. С. 9-14.

### И.А. ПОЛИЛОВ

Рязанский государственный радиотехнический университет

### КОМПЛЕКСНЫЕ СИСТЕМЫ БЕЗОПАСНОСТИ ДЛЯ ПРОМЫШЛЕННЫХ ПОМЕЩЕНИЙ

*В статье рассматриваются комплексные системы безопасности для промышленных помещений.*

В современной промышленности представлено множество высокотехнологичных решений, обеспечивающих комплексную защиту различных помещений. Наиболее эффективными являются комплексные интегрированные системы безопасности. Они производят мониторинг работы подсистем, записывают информацию для анализа, управляют и обеспечивают взаимодействие между подсистемами в автоматическом режиме. Комплексные системы безопасности решают важнейшие задачи на предприятиях. Они обеспечивают защиту помещений от пожаров, проникновения посторонних и выполняют автоматизацию различных процессов, что позволяет существенно улучшить эффективность работы производства.

К интегрированным системам безопасности относят системы охранной и пожарной сигнализации, видеонаблюдения, системы контроля и управления доступом. Современные инженерные возможности позволяют при необходимости дополнить системы дополнительными подсистемами. В качестве примера можно привести такие комплексные системы безопасности как Орион, PERCo, Бастион.

Такие комплексы состоят из совокупности подсистем: охранной сигнализации, контроля доступа, видеонаблюдения, пожарной сигнализации, пожаротушения, дымоудаления, систем защиты персональных данных, систем проводной и радиосвязи, речевого оповещения и трансляции.

Системы охранной сигнализации обнаруживают несанкционированное проникновение на охраняемый объект, оповещают об этом факте службы охраны, ведомственные органы и/или владельца помещения непосредственно (в зависимости от настроек и типа охраны), фиксирует точные дату и время попытки проникновения, определяют рубеж охраны, который был нарушен.

Системы отражения информации о характере сигнала от датчиков должны иметь как минимум три варианта «взят под охрану», «снят с охраны», «неисправен». Выпускаемые сегодня приборы приемно-контрольные (ППКО) для охранной сигнализации способны работать с огромным количеством разных охранных датчиков, поэтому выбор датчиков зависит от конкретного помещения и задач которые датчик должен выполнять.

Пожарная сигнализация определяет возгорания на начальной стадии и своевременно оповещает об этом заинтересованных лиц. Необходимость установки систем такого рода, во многих случаях обусловлена не только заботой владельца объекта о собственной безопасности и сохранности имущества, но и нормативными актами. Системы пожарной сигнализации также используются для формирования сигнала на запуск автоматических устройств пожаротушения, дымоудаления, оповещения о пожаре, а также управления технологическими, электротехническими и прочими видами оборудования.

Типы современных систем пожаротушения делятся на водные, пенные, аэрозольные, порошковые, газовые. Согласно требованиям нормативной документации, тип установки системы пожаротушения, способ тушения и огнетушащее вещество для каждого конкретного объекта определяется с учетом пожарной опасности и физико-химических свойств производимых, хранимых и применяемых веществ и материалов, а также особенностей защищаемого оборудования и строительных конструкций.

Датчики дыма предназначены для обнаружения возгораний, сопровождающихся выделением дыма в помещениях. На данный момент, это самый распространенный вид датчиков пожара устанавливаемых в помещениях. Характеризуется высокой обнаруживающей способностью, на ранней стадии возгорания. Датчики дыма, устанавливаются на потолке, либо в межпотолочном пространстве, между основным и подвесным потолком. Количество датчиков дыма, определяется исходя из площади помещения и высоты потолка. При высоте потолка до 3,5 метров, один датчик дыма, может контролировать объем до 80 кв. м. площади. Согласно правилам, в любом помещении, независимо от его площади, должно присутствовать не менее двух датчиков.

Системы контроля и управления доступом (СКУД) предназначены для организации санкционированного доступа на территорию и во внутренние помещения. Обычно СКУД используются как одна из систем интегрированной системы безопасности. Наиболее распространенная интеграция - с системами видеонаблюдения и системой охранной сигнализации.

По типу различают сетевые и автономные системы. Сетевые системы включают в свой состав один или несколько компьютеров. С их помощью ведется мониторинг происходящего на объекте и управление настройками составляющих системы. Автономные системы не имеют предусмотренных каналов связи с компьютером и управляются с помощью переключателей либо мастер-карт. При оснащении средних и крупных помещений автономные системы используются крайне редко.

Основная отличительная черта сетевых СКУД – возможность их конфигурирования с ПК. Сетевые СКУД имеют свои особенности, они отличаются по архитектуре, масштабу, типу считывателей, степени устойчивости к взлому. Главный элемент сетевых СКУД – программное обеспечение.

Управляющий контроллер СКУД обычно представляет собой устройство на базе промышленного персонального компьютера с операционной системой Linux. Плюсы такой реализации СКУД – удобство программирования, наглядное отображение событий в режиме реального времени, возможность построения систем большого масштаба, а также возможность интеграции с различными системами безопасности. К минусам относят обучение персонала, более высокую стоимость приобретения и монтажа по сравнению с автономными системами.

Системы обеспечения комплексной безопасности объекта должны быть сформированы для эффективного противостояния перечис-

ленным угрозам путем принятия адекватных мер защиты. Для этого должны быть созданы и надежно функционировать следующие подсистемы безопасности: подсистема защиты объекта при чрезвычайных ситуациях, подсистема физической защиты (охранной безопасности), подсистема информационной безопасности.

Обеспечение эффективной безопасности предполагает решение проблем моделирования угроз, их качественной и количественной оценок с учетом сложности структурно-функционального построения системы безопасности, ее элементов, а также данных о внешних воздействиях естественного и искусственного происхождения.

**Г.А. ПОПОВ, С.В. СКВОРЦОВ**

Рязанский государственный радиотехнический университет

### **АНАЛИЗ ВОЗМОЖНОСТИ ПАРАЛЛЕЛЬНОГО РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА**

*Рассматриваются подходы к распараллеливанию известных методов решения задачи коммивояжера средствами персональных ЭВМ.*

Задача коммивояжера заключается в отыскании гамильтонова цикла минимального веса на полном взвешенном графе. Доказано, что задача коммивояжера является NP-трудной задачей. Класс NP – это множество задач, время решения которых сильно зависит от размерности входных данных. В частности, размер пространства решений зависит экспоненциально от количества вершин графа. Это требует ускорения процессов решения данной задачи, одним из способов достижения которого является распараллеливание алгоритмов [1, 2].

В данный момент распараллеливание возможно даже на базе персональных ЭВМ, которые являются дешевыми и повсеместно распространены. Для этого могут использоваться два подхода к построению многопоточных программ - на многоядерном процессоре и на графическом ускорителе. В первом случае обычно распараллеливают поток логики [3], например, отделяют друг от друга участки кода [4, 5], которые выполняют каждый свою задачу, во втором случае распараллеливают обработку независимых многомерных данных [6]. В зависимости от алгоритма преимущество дает один или другой способ.

Целью работы является анализ возможных подходов к распараллеливанию известных методов решения задачи коммивояжера.

Все методы можно разделить на простейшие, жадные, эвристические. Все эффективные методы (сокращающие полный перебор) -

методы эвристические. В большинстве эвристических методов находится не самый эффективный маршрут, а приближённое решение. Каждый из алгоритмов с тем или иным успехом можно распараллелить.

Рассмотрим несколько наиболее известных методов решения задачи коммивояжера и возможные способы их распараллеливания.

**Полный перебор.** Это простейший метод решения задачи коммивояжера, который основан на переборе каждого возможного решения и выборе самого лучшего.

Первый способ распараллеливания – разделить все множество решений на несколько подмножеств, после чего выполнить поиск оптимального решения в этих подмножествах в разных потоках.

Второй способ – модифицировать метод - объединить узлы в группы. Теперь можно смотреть на задачу полного перебора как на рекурсивную. И каждый отдельный участок данного алгоритма можно выполнить в своем потоке.

**Метод Монте-Карло.** Решение задачи коммивояжера методом перебора может быть немного изменено, чтобы обеспечить возможность если и не получить оптимальный маршрут, то найти какой-то не самый плохой вариант решения. Здесь мы можем перебирать не все варианты, а случайно созданные неким механизмом.

Для распараллеливания можно запустить данный алгоритм на нескольких ядрах процессора, а при истечении некоторого времени выбирать лучший из полученных каждым потоком.

Поскольку этот метод может работать при большом количестве вершин графа, то удобно вычислять функцию пути на графическом ускорителе.

**Метод ветвей и границ** – это один из немногих алгоритмов, которые дают оптимальное решение задачи коммивояжера. Является вариантом поиска с возвратом, который позволяет найти точное решение. В основе метода ветвей и границ лежит идея последовательного разбиения множества допустимых решений на два подмножества — содержащее выбранное ребро и не содержащее.

Метод ветвей и границ плохо поддается распараллеливанию в связи с его непредсказуемостью во время построения дерева решений.

Первый способ – параллельное решение подзадач, возникающих при разбиении множества решений исходной задачи. Данный способ не позволяет получить постоянный выигрыш в связи необходимостью возвратов к ранее полученным подзадачам.

Второй способ – параллельная обработка элементов матрицы расстояния с помощью графического ускорителя. Данный способ не



даст выигрыша в производительности, так как метод ветвей и границ не может работать с задачами большой размерности.

Третий способ – параллельное вычисление оценок целевой функции для двух новых вершин дерева поиска, получаемых при его ветвлении (разбиение множества решений на два подмножества). Данные вычисления можно выполнять параллельно в отдельных потоках, никак не беспокоясь о синхронизации.

**Генетический алгоритм.** В качестве функции приспособленности используется длина пути коммивояжера. Для ускорения алгоритма целесообразно параллельно запускать работу над несколькими популяциями, после чего через заранее заданный период времени они объединяются. Данный способ поможет избежать локальных минимумов, в которые может попасть «главная» популяция.

Еще один вариант - это параллельная обработка элементов матрицы расстояний с помощью графического ускорителя. Так как генетический алгоритм может работать с большим количеством вершин графа, то данный способ будет давать явный выигрыш при большой размерности задачи.

Возможен и третий способ, который является объединением первого и второго вариантов. Однако при его программной реализации возрастает сложность синхронизации потоков.

Таким образом, проведенный анализ показывает целесообразность программной реализации и дальнейшего исследования следующих методов и подходов к их распараллеливанию: метод Монте-Карло с параллельным запуском нескольких экземпляров алгоритма в разных потоках; метод ветвей и границ с параллельным вычислением оценок целевой функции для пары подмножеств решений, получаемых при каждом ветвлении дерева поиска; генетический алгоритм с параллельной обработкой нескольких популяций и периодическим их объединением для создания новых популяций.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Корячко В.П., Скворцов С.В. Иерархическая модель глобальной оптимизации параллельных объектных программ // Информационные технологии. 1998. № 9. С. 34-40.
2. Першин А.С., Скворцов С.В. Распределение регистровой памяти в системах параллельной обработки данных // Системы управления и информационные технологии. 2007. Т. 27. № 1. С. 65-70.

3. Скворцов С.В. Алгоритм планирования параллельных вычислений в многоядерных процессорах // Радиотехника. 2016. № 8. С. 153-159.

4. Скворцов С.В. Целочисленные модели оптимизации кода по критерию времени // Информационные технологии. 1997. № 10. С. 2-7.

5. Скворцов С.В. Оптимизация кода для суперскалярных процессоров с использованием дизъюнктивных графов // Программирование. 1996. № 2. С. 41-52.

6. Лунин Д.В., Скворцов С.В. Разработка параллельного генетического алгоритма для решения задачи коммивояжера на платформе CUDA // Системы управления и информационные технологии. 2015. Т. 60. № 2. С. 50-55.

### **С.В. ПОПОВ**

Рязанский государственный радиотехнический университет

## **КОНСТРУИРОВАНИЕ БЛОКА УПРАВЛЕНИЯ SDN СЕТИ И РАЗРАБОТКА АЛГОРИТМОВ МНОГОПУТЕВОЙ МАРШРУТИЗАЦИИ**

*В статье рассматривается один из алгоритмов маршрутизации, применяемый в SDNсетях, а так же представлена информация о проверке алгоритмов.*

Значительный недостаток сетевой инфраструктуры заключается в том, что каждое сетевое устройство снабжено собственным элементом управления. При формировании маршрута, информация будет обрабатываться в каждом коммутаторе сети.

Сейчас активно развиваются программно-конфигурируемые сети (SoftwareDefinedNetworks) – это сети, в которых уровень управления отделен от уровня передачи данных. Данная технология позволяет заметно повысить эффективность оборудования, при этом снизить эксплуатационные затраты. Особенности SDN сетей заключается в следующем:

- разделение процессов передачи и управления данными;
- унифицированный интерфейс для управления и передачи данных;
- централизованное управление сетью с помощью ОС контроллера.

Для организации централизованного управления контроллер должен содержать в себе алгоритмы маршрутизации. Реализация этих алгоритмов возможна, если он обладает информацией о параметрах сети, таких как задержка (delay), стоимость (cost) и пропускная способность (bandwidth). Поэтому, всегда ставится задача снабжения управляющего устройства сети актуальными данными.

Не смотря на огромное разнообразие существующих алгоритмов маршрутизации, тенденция к разработке новых отмечается и сегодня. Для проверки “свежих” алгоритмов существуют специальные эмуляторы компьютерных сетей. Одним из таких эмуляторов является mininet. Он позволяет реализовать компьютерную сеть, состоящую из хостов, коммутаторов и OpenFlow контроллеров.

Проверка алгоритма маршрутизации заключается в исследовании его эффективности и определении области применения. Для этого формируется топология, в которой производится несколько запусков алгоритма. Перед новым пуском производится изменение параметров сети. Далее происходит то же самое, но для новой топологии. По результатам проверок можно судить об эффективности алгоритма, сравнивая его с распространенными алгоритмами маршрутизации.

Многопутевая маршрутизация подразумевает поиск нескольких маршрутов следования от источника к узлу назначения. Она находит широкое применение на практике, например, передача данных сразу по нескольким маршрутам, или формирование резервных на случай обрыва основного соединения. Обычно количество искомых путей ограничивают, для увеличения скорости работы алгоритма. К многопутевым алгоритмам можно отнести:

- алгоритм Йена;
- алгоритм MCP;
- алгоритм LARAC.

Рассмотрим алгоритм MCP. Он разрешает задачу многопутевого поиска при наличии нескольких ограничений. Алгоритм выбирает только те маршруты, параметры которых не выходят за заданные пределы. Как и остальные многопутевые алгоритмы, он основан на сведении поставленной задачи к одной метрике, так как не существует методов динамического программирования для разрешения многокритериальных задач.

Весь алгоритм можно разбить на три шага:

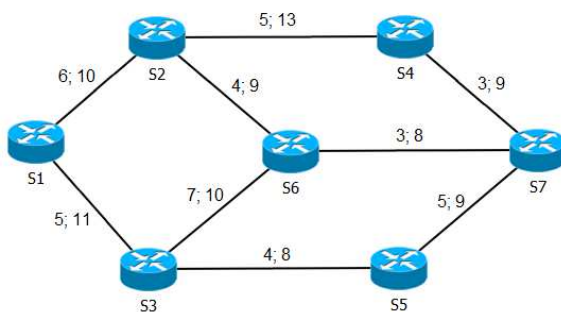
**Шаг 1.** Первым шагом алгоритма будет поиск всех возможных маршрутов между источником и узлом назначения, с помощью алгоритма Йена.

**Шаг 2.** Теперь необходимо выбрать метрику, которую мы будем рассматривать первой, например задержку. Для каждого маршрута, найденного с помощью алгоритма Йена, производится расчет параметра задержки. Если значение задержки не превышает заданного ограничения, то маршрут остается в рассмотрении, иначе исключается.

**Шаг 3.** Далее аналогичные действия производятся для всех маршрутов, оставшихся в рассмотрении после шага 2, по параметру стоимости.

Таким образом, мы получим несколько маршрутов, параметры которых удовлетворяют заданным ограничениям. В том случае если нет маршрутов, удовлетворяющих ограничениям, выбирается один маршрут, для которого, значение по одной из рассматриваемых метрик является минимальным. Стоит отметить, что за первую метрику обычно принимают более жесткий параметр сети, который исключает большее число маршрутов, что уменьшает трудоемкость алгоритма на третьем шаге.

Для демонстрации работы алгоритма МСР приведем простой пример. Пусть граф сети состоит из семи коммутаторов. Каждое ребро описано двумя параметрами: задержки и стоимость (Рис. 1). Необходимо найти все маршруты из коммутатора 1 в коммутатор 7, удовлетворяющие ограничениям: по задержке 14мс, по стоимости 30.



**Рис. 1 – Граф сети.**

На рис. 1 изображены метрики следующим образом: задержка; стоимость.

На первом шаге получим следующий набор маршрутов:

- 1) S1 → S2 → S4 → S7 (14; 32);
- 2) S1 → S2 → S6 → S7 (13; 27);
- 3) S1 → S3 → S6 → S7 (15; 29);
- 4) S1 → S3 → S5 → S7 (14; 28).

После второго шага рассматривая метрику задержки:

- 1)  $S1 \rightarrow S2 \rightarrow S4 \rightarrow S7$  (14; 32);
- 2)  $S1 \rightarrow S2 \rightarrow S6 \rightarrow S7$  (13; 27);
- 3)  $S1 \rightarrow S3 \rightarrow S5 \rightarrow S7$  (14; 28).

После третьего шага рассматривая метрику стоимости:

- 1)  $S1 \rightarrow S2 \rightarrow S6 \rightarrow S7$  (13; 27);
- 2)  $S1 \rightarrow S3 \rightarrow S5 \rightarrow S7$  (14; 28).

Таким образом, из всех имеющихся маршрутов из коммутатора 1 до коммутатора 7, всего два удовлетворяют заданным ограничениям. Очевидно, что маршруты, которые можно было проложить через большее число коммутаторов, не прошли по заданным ограничениям (например,  $S1 \rightarrow S2 \rightarrow S6 \rightarrow S3 \rightarrow S5 \rightarrow S7$ ).

В простейшем случае топология сети может состоять из одного коммутатора, к которому подключено несколько компьютеров. В данном случае, реализация мультитепловой маршрутизации не возможна, так как имеется единственный маршрут. Аналогичная ситуация происходит в сети, граф которой представляет собой дерево (Рис. 2). В данном, случае проверяется адаптивность тестируемого алгоритма к малым сетям, а также к нестандартным сетям.

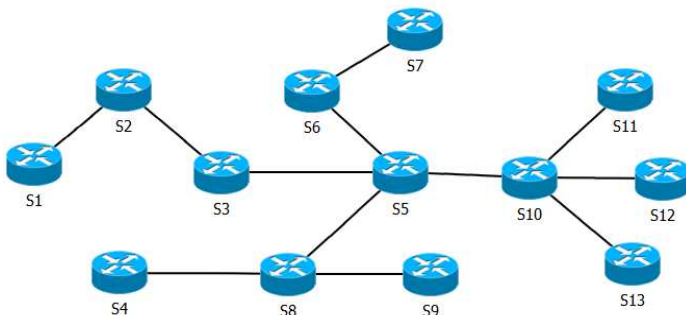


Рис. 2 – Граф-дерево сети.

Неотъемлемым этапом проверки эффективности алгоритмов маршрутизации является тестирование на реальной сети. Для этого необходимо собрать аппаратный стенд. Оптимальным решением, с точки зрения компактности стенда, является установка всех коммутаторов сети в единую стойку, что позволяет быстро перестроить тестируемую топологию сети.

Таким образом, проверку эффективности алгоритмов маршрутизации можно разбить на два крупных этапа. Первым этапом является отладка работы алгоритма в эмуляторе компьютерных сетей. Так можно выявить область применения, а также ряд ошибок, возникающих

при работе алгоритмов. Вторым этапом является тестирование на реальной аппаратуре. Это позволяет устранить недостатки в плане совместимости программного кода с аппаратным продуктом, для которого он предназначен. Такой метод тестирования алгоритмов маршрутизации позволяет выявить их слабые стороны, до внедрения в корпоративные сети.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Смелянский Р.Л. Программно-конфигурируемые сети. Журнал "Открытые системы. СУБД", ноябрь 2012, №09.
2. William Stallings. "Software-Defined Networks and OpenFlow", интернетжурнал, март 2013, 16 стр, №1.
3. «Software-defined networking: the new norm for Networks», ONF White Paper. April 2012.

### **Т.А. ПЮРОВА, С.В. СКВОРЦОВ**

Рязанский государственный радиотехнический университет

### **РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ПОИСКА ДАННЫХ НА ПЛАТФОРМЕ CUDA**

*В данной статье рассматривается ряд вопросов, связанных с разработкой, реализацией и анализом эффективности параллельных алгоритмов поиска данных, основанных на технологии CUDA.*

В последние годы программирование является не только средством принятия решений для результативной работы в различных прикладных областях, но и областью науки. Очевидно, что использование алгоритмов играет важную роль при решении различных проблем с данными. Изо дня в день растет поток данных, работа с которыми занимает большое количество времени и требует немалых трудозатрат. Поэтому одним из основных действий со структурированной информацией является поиск.

Поиск – процедура нахождения определенной информации во множестве данных, представляющие собой массивы, списки, записи, каждая из которых имеют один ключ или больше. Ключ поиска – это поле, на основании значения которого осуществляется поиск. Цель поиска заключается в нахождении элементов, записей, удовлетворяющих значению ключа, или их отсутствию.

В виду того, что поиск является очень распространенным действием в программировании, то возникает потребность в его оптимизации. Одним из возможных способов является распараллеливание, то есть параллельное выполнение одинаковых операций над частями массива, списка, записей. Существует множество технологий, основанных на дроблении последовательности данных, например, OpenMP, OpenCL, CUDA[1]. Последняя технология представляет особый интерес, так как она задействует графический процессор – видеокарту фирмы NVIDIA[2].

В настоящее время выбор алгоритмов поиска зависит от способа организации данных. У любого алгоритма поиска есть преимущества и недостатки. Поэтому выбор алгоритма должен быть основан на наилучшем подходе для решения конкретной задачи.

Главным образом, идеи всех алгоритмов поиска сосредоточены в стратегии поиска и методах перебора. Рассмотрим бинарный и интерполяционный алгоритмы поиска более подробно.

Бинарный поиск – это нахождение заданного элемента на упорядоченном множестве, которое осуществляется путем многократного деления этого множества на две равные части таким образом, что искомый элемент оказывается в одной из этих частей. Процедура поиска завершается при совпадении нужного элемента с элементом, являющимся границей между подмножествами или при его отсутствии.

Идея данного метода заключается в следующем. Нахождение ключа поиска среди элементов упорядоченного массива начинается с определения центрального элемента. Значение выбранного элемента сравнивается с искомым, и если значения данных элементов оказываются равны, то поиск успешно завершается. Если ключ больше центрального или меньше, то выделяется подмассив элементов, находящихся справа или слева от центрального соответственно, и процедура поиска повторяется в сформированном подмассиве.

Также в упорядоченном массиве может встречаться несколько элементов, значение которых равны ключу. Тогда результатом работы алгоритма поиска будет первый совпавший с ключом элемент.

Схема реализации алгоритма бинарного поиска представлена в таблице 1, где в качестве ключа выбирается элемент со значением 12. Курсивом отмечен ключ поиска, а сравниваемый элемент выделен полужирным шрифтом.

Таблица 1 – Бинарный поиск.

| Шаг | Элементы массива |   |   |   |   |   |           |           |    |
|-----|------------------|---|---|---|---|---|-----------|-----------|----|
|     | 1                | 2 | 4 | 5 | 7 | 9 | 10        | 12        | 17 |
| 1   | 1                | 2 | 4 | 5 | 7 | 9 | 10        | 12        | 17 |
| 2   |                  |   |   |   |   | 9 | <b>10</b> | 12        | 17 |
| 3   |                  |   |   |   |   |   |           | <b>12</b> | 17 |

Идея распараллеливания алгоритма бинарного поиска заключается в том, чтобы изначально запустить на параллельное выполнение как можно большее количество потоков[3,4], в которых будет осуществляться поиск. Технологически исходный массив разбивается на блоки, в каждый из которых размещается часть массива, поиск в которой осуществляется независимо от других таких частей средствами графического ускорителя. Таблица 2 наглядно иллюстрирует данный процесс, причем после каждого разделения некоторой части массива степень параллелизма (число параллельных потоков) уменьшается.

Таблица 2 – Параллельный бинарный поиск.

| Этап<br>число потоков) | Элементы массива |          |   |   |   |   |    |           |    |
|------------------------|------------------|----------|---|---|---|---|----|-----------|----|
|                        | 1                | 2        | 4 | 5 | 7 | 9 | 10 | 12        | 17 |
| Этап 1<br>(3 потока)   | 1                | <b>2</b> | 4 |   |   |   |    |           |    |
|                        |                  |          |   | 5 | 7 | 9 |    |           |    |
|                        |                  |          |   |   |   |   | 10 | <b>12</b> | 17 |

Функция ядра данного алгоритма Kernel имеет следующий вид.

```

__global__ void Kernel(float *data, int k, int *result, int x)
{
    int x = 16 * 1024 * 1024; // количество элементов
    int low = 0; // нижняя граница
    int up = x-1; // верхняя граница
    int step; // шаг
    int idx; // текущий поток
    while (up - low > k) // пока количество элементов между границами больше количества потоков (в данном случае потоков k)
    {
        step = (up - low) / k;
        idx = low + (blockIdx.x * blockDim.x + threadIdx.x) * step;

        if ( data[idx] == x ) // найден искомый элемент
        {
            result[0] = data [idx];
        }
    }
}

```



```

    result[1] = idx; break;
  } else if ( data[idx] > x )
  {
    if(data[idx] < up)
      up = data[idx]; //минимальный среди больших
  } else if(data[idx] < x)
  {
    if(data[idx] > low)
      low = data[idx]; // максимальный среди меньших
  }
}
}
}

```

Достоинство данного алгоритма заключается в относительной скорости поиска, по сравнению с линейным алгоритмом, где поиск осуществляется путем перебора каждого значения. Недостатком является его применение только на упорядоченном множестве.

Интерполяционный поиск связан непосредственно с понятием «интерполяция» - процесс нахождения неизвестных значений на основе имеющихся, то есть, используя индексы известных значений элементов в массиве, можно определить положение искомого элемента.

Если выбранная при помощи интерполяции позиция оказывается ошибочной, то алгоритм сравнивает ключ поиска со значением выбранного элемента. Если значение искомого элемента меньше значения сравниваемого элемента, то поиск продолжается в левой части массива, если больше — в правой части.

При бинарном поиске массив разбивается на две равные части. Интерполяционный поиск работает по-другому: на каждой итерации происходит разбиение массива таким образом, чтобы найти ближайший элемент к ключу, причем точка разбиения определяется по следующей формуле:

$$middle = left + (key - mas [left ]) * (right - left ) / (mas [right ] - mas [left ]), \quad (1)$$

где *middle* – элемент, сравниваемый с ключом; *left* – индекс левой границы поиска; *right* – индекс правой границы поиска; *mas[ ]* – исходный массив поиска.

Схема реализации алгоритма интерполяционного поиска представлена в таблице 3.

**Таблица 3 – Интерполяционный поиск.**

| Шаг | Элементы массива |   |   |   |   |          |    |           |    |
|-----|------------------|---|---|---|---|----------|----|-----------|----|
|     | 1                | 2 | 4 | 5 | 7 | 9        | 10 | 12        | 17 |
| 1   | 1                | 2 | 4 | 5 | 7 | <b>9</b> | 10 | 12        | 17 |
| 2   |                  |   |   |   |   |          | 10 | <b>12</b> | 17 |

Идея параллельной многопоточной реализации алгоритма интерполяционного поиска иллюстрируется в таблице 4, где предполагается использование нисходящего распараллеливания[3]. Каждая строка соответствует подмассиву элементов, в котором поиск осуществляется в потоке независимо от других потоков.

**Таблица 4 – Параллельный интерполяционный поиск.**

| Этап<br>число потоков) | Элементы массива |          |          |          |          |          |           |           |    |           |
|------------------------|------------------|----------|----------|----------|----------|----------|-----------|-----------|----|-----------|
|                        | 1                | 2        | 4        | 5        | 7        | 9        | 10        | 12        | 17 |           |
| Этап 1<br>(9 потоков)  | <b>1</b>         |          |          |          |          |          |           |           |    |           |
|                        |                  | <b>2</b> |          |          |          |          |           |           |    |           |
|                        |                  |          | <b>4</b> |          |          |          |           |           |    |           |
|                        |                  |          |          | <b>5</b> |          |          |           |           |    |           |
|                        |                  |          |          |          | <b>7</b> |          |           |           |    |           |
|                        |                  |          |          |          |          | <b>9</b> |           |           |    |           |
|                        |                  |          |          |          |          |          | <b>10</b> |           |    |           |
|                        |                  |          |          |          |          |          |           | <b>12</b> |    |           |
|                        |                  |          |          |          |          |          |           |           |    | <b>17</b> |

Ядро данного алгоритма поиска представлено ниже.

```

__global__ void Kernel ( float * data, int n, int find, int result )
{
    int idx; // текущий поток
    int low =0; // нижняя граница
    int up = n; // верхняя граница
    // пока искомый элемент не найден или пределы поиска еще
    существуют
    while ((data[low] < find) && (data[up] > find))
    {
        //вычисление интерполяцией следующего элемента, который
        будет сравниваться с искомым
        idx = blockIdx.x + ((find - data[blockIdx.x]) * (blockIdx.x *
        blockDim.x + threadIdx.x - blockIdx.x)) / (data[blockIdx.x * blockDim.x
        + threadIdx.x] - data[blockIdx.x]);
        //Получение новых границ области, если искомый элемент не
        найден
        if (data[idx] < find) low = idx + 1;
        else if (data[idx] > find) up = idx - 1;
        else result=idx;
        //Если искомый элемент найден на границах области поиска
        if (data[low] == find) result=low;
    }
}

```

```
else if (data[up] == find) rezult=up;  
else result=-1;  
}}
```

Главное преимущество данного алгоритма заключается в скорости его выполнения поиска, причем он осуществляется быстрее бинарного алгоритма, так как область поиска сокращается неравномерно. Основным недостатком также является обязательное условие упорядоченности исходного массива данных.

Теоретически сложность рассматриваемых алгоритмов поиска подтверждает превосходство интерполяционного поиска над бинарным. В среднем сложность бинарного алгоритма пропорциональна  $O(\log n)$ , а интерполяционного –  $O(\log(\log n))$ . Результаты практических испытаний показали, что даже при малых размерах массива данных бинарный алгоритм выполняет поиск медленнее, чем интерполяционный, в несколько раз.

Таким образом, использование разных алгоритмов поиска, реализованных в данной работе, позволяет не только значительно сократить время, затраченное на трудоемкие операции поиска данных, но и в конечном итоге повысить производительность труда.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Скворцов С.В. Алгоритм планирования параллельных вычислений в многоядерных процессорах // Радиотехника. - 2016. - № 8. - С. 153-159.
2. Першин А.С., Скворцов С.В. Распределение регистровой памяти в системах параллельной обработки данных // Системы управления и информационные технологии. 2007. Т. 27. № 1. С. 65-70.
3. Скворцов С.В., Пюрова Т.А. Параллельные алгоритмы сортировки данных и их реализация на платформе CUDA // Вестник Рязанского государственного радиотехнического университета. - 2016. - № 58. - С. 42-48.
4. Лунин Д.В., Скворцов С.В. Разработка параллельного генетического алгоритма для решения задачи коммивояжера на платформе CUDA // Системы управления и информационные технологии. 2015. Т. 60. № 2. С. 50-55.

**В.Н. РУЧКИН, Е.С. ГЕРАЩЕНКО, В.Ю. ПОТАПОВА,  
А.С. ТАРАСОВ, А.В. ТОКАРЕВ**

Рязанский государственный радиотехнический университет

## **ШИФРОВАНИЕ И ДЕШИФРОВАНИЕ ИНФОРМАЦИИ НА НЕЙРОПРОЦЕССОРЕ NM6403**

*Рассматривается программирование алгоритма ГОСТ 28147-89 в режиме простой замены для нейропроцессора NeuroMatrix NM6403.*

Шифрование и дешифрование информации являются важными прикладными задачами, что связано с необходимостью обеспечения безопасной передачи данных по каналам связи.

Существует большое количество алгоритмов шифрования, среди которых можно выделить IDEA, DES, ГОСТ 28147-89. Алгоритм IDEA использует 128-битный ключ и 64-битный размер блока. Исходное сообщение разбивается на блоки по 64 бит, в том случае, если подобное разбиение невозможно, исходное сообщение добавляются нулевые биты [1]. Данный алгоритм использует математические операции, такие как сложение по модулю 2, умножение по модулю и побитовое исключение ИЛИ (XOR). Алгоритм DES использует 56-битовый ключ и 64-размер блока.

Недостатком вышеизложенных методов являются большие вычислительные затраты на реализацию. Алгоритм ГОСТ 28147-89 значительно проще в реализации, и хотя по некоторым критериям он уступает другим методам, он наилучшим образом подходит для шифрования небольших блоков информации, например, ключевых данных. В этих случаях нет необходимости применять сложные многоступенчатые алгоритмы. Кроме того, за счет простоты ГОСТ 28147-89 он легко переносится на аппаратную основу.

В данной работе шифрование данных осуществляется для архитектуры нейропроцессора NeuroMatrix NM6403, микропроцессорная архитектура которого разработана в компании НТЦ «Модуль». Устройства, которые строятся на такой аппаратной структуре, предназначены для цифровой обработки сигналов и создания нейронных сетей. В состав архитектуры входят устройства управления, а также скалярные и векторные узлы. Имеется возможность работы с внешней памятью с использованием программируемых интерфейсов различного типа. Допустима интеграция нейропроцессора в микропроцессорные системы [2].

Алгоритм ГОСТ 28147-89 представляет собой стандарт симметричного блочного шифрования, принятый в 1989 году и объявленный полностью открытым в 1994 году.

Существуют 4 режима работы алгоритма:

- режим простой замены;
- гаммирование;
- гаммирование с обратной связью;
- режим выработки имитовставки.

В данной работе используется режим простой замены, который по-другому называется режимом электронной кодовой книги. Основная особенность метода заключается в том, что каждый блок открытого текста заменяется блоком шифротекста по определенному принципу. Существует ограничение на длину слова, равную 64 битам, однако оно не влияет на возможность шифрования ключевых данных.

На основе вышеизложенного алгоритма была реализована программа, в которой в качестве исходных данных используется 32-разрядное слово. Ключ содержит 48 разрядов, при этом в каждом шаге алгоритма используется очередная 16-битная часть ключа. Основным шагом алгоритма выполняется три раза.

Словесное описание основного шага алгоритма шифрования:

1. 32-разрядное слово разделяется на две 16-разрядные части N1 и N2 (старшая и младшая).
2. Старшая часть N1 складывается по модулю два с используемой на этом шаге частью ключа.
3. Осуществляется поблочная замена по таблице результата предыдущего шага.
4. Выполняется циклический сдвиг влево на 11 бит результата предыдущего шага.
5. Сдвинутое значение складывается по модулю два с младшей частью кодируемого слова.
6. Формируется итоговое слово шага алгоритма: в качестве его старшей части используется младшая часть исходного слова, а в качестве младшей – результат предыдущего шага.

В таблице 1 приведены значения для поблочной замены.

Алгоритм дешифрации также реализован по алгоритму ГОСТ 28147-89.

Таблица 1 – Значения для поблочной замены.

|   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 13 | 8  | 5  | 10 | 7  | 2  | 11 | 4  | 1  | 15 | 3  | 6  | 14 | 0  | 12 | 9  |
| 1 | 4  | 7  | 8  | 14 | 0  | 11 | 15 | 1  | 2  | 13 | 12 | 5  | 10 | 9  | 3  | 6  |
| 2 | 10 | 11 | 14 | 6  | 9  | 13 | 3  | 8  | 7  | 4  | 0  | 1  | 15 | 5  | 2  | 12 |
| 3 | 15 | 13 | 8  | 7  | 11 | 14 | 12 | 3  | 2  | 10 | 9  | 4  | 6  | 1  | 5  | 0  |
| 4 | 6  | 14 | 12 | 5  | 10 | 3  | 0  | 11 | 15 | 3  | 4  | 7  | 2  | 9  | 1  | 8  |
| 5 | 11 | 4  | 6  | 15 | 1  | 13 | 2  | 9  | 14 | 7  | 12 | 0  | 10 | 5  | 8  | 3  |
| 6 | 8  | 15 | 9  | 6  | 12 | 7  | 3  | 5  | 0  | 2  | 14 | 13 | 4  | 11 | 10 | 1  |
| 7 | 14 | 9  | 15 | 13 | 8  | 10 | 11 | 12 | 4  | 1  | 5  | 6  | 7  | 3  | 0  | 2  |

В качестве иллюстрации работы программы рассмотрим шифрование и дешифрование по алгоритму ГОСТ 28147-89. В качестве исходного слова выберем HYDROGEN, ключа - ABCD EFGH KLMN. Буквенное представление переводится в коды ASCII, по которым производится операции шифрования и дешифрования.

В таблице 2 представлено исходное и зашифрованное слово.

Таблица 2 – Исходное и зашифрованное слово.

| Н  | У  | Д  | Р  | О  | Г  | Е  | Н  |
|----|----|----|----|----|----|----|----|
| 48 | 59 | 44 | 52 | 4F | 47 | 45 | 4E |
| 87 | 3C | C5 | 46 | 5B | A9 | DC | CA |

На рис. 1-2 представлено представление результатов шифрования и дешифрования в памяти эмулятора.

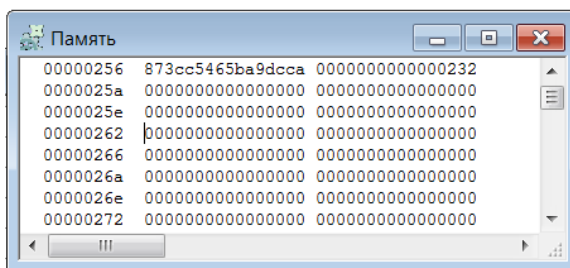


Рис. 1 – Зашифрованное слово в памяти.

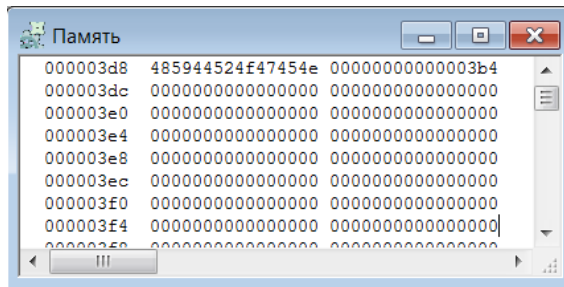


Рис. 2 – Дешифрованное слово в памяти.

Представленный алгоритм шифрования на нейропроцессоре NM6403 имеет достаточно простой криптошифр, позволяющий с минимальными аппаратными затратами зашифровать необходимое сообщение. Основной недостаток метода состоит в однозначности соответствия блока исходного текста и его зашифрованного варианта, так как табличная замена осуществляется по одной и той же таблице. Однако если периодически изменять шифросхемы, то алгоритм становится практически полностью защищённым от атак типа man-in-middle [3,4]. Поэтому к основным областям применения подобных алгоритмов можно отнести передачу данных между относительно близко расположенными объектами, при которых возможно постоянно выполнять обмен шифрограммами с некоторыми заданными интервалами.

Если усилить данный алгоритм какими-либо ещё системами шифрования, то алгоритм может быть применён и во множестве различных видов задач, включая и военные.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Конкин Ю.В., Гринченко Н.Н., Кафтаева М.Ю. Модуль определения и хранения значений дешифровочных признаков объектов заданного класса на основе прямых и косвенных показателей. Хроники объединенного фонда электронных ресурсов Наука и образование. 2013. № 7 (50). С. 8.
2. Костров Б.В., Ручкин В.Н. Архитектура микропроцессорных систем. Учебное пособие для студентов, обучающихся по специальности 230101 "Вычислительные машины, комплексы, системы и сети" / Б. В. Костров, В. Н. Ручкин. Москва, 2007.
3. Злобин В. К., Ручкин В. Н. Нейросети и нейрокомпьютеры. Санкт-Петербург, 2011.
4. Костров Б.В., Гринченко Н.Н., Геращенко Е.С., Потапова В.Ю., Тарасов А.С., Токарев А.В. Моделирование канала передачи

аэрокосмических изображений с использованием каскадного кода. Известия Тульского государственного университета. Технические науки. 2016. № 2. С. 124-129.

**А.Н. САПРЫКИН, К.Д. АКИНИНА**

Рязанский государственный радиотехнический университет

### **ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА КОМПОНОВКИ БЛОКОВ ЭЛЕКТРОННО- ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ**

*Рассматриваются основные характеристики генетического алгоритма компоновки блоков электронно-вычислительных средств. Описываются используемые операторы генетического поиска.*

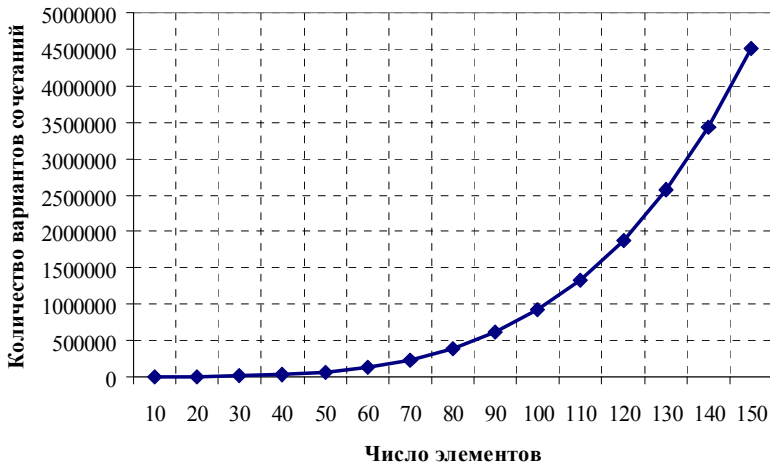
Компоновка конструктивных модулей электронно-вычислительных средств является одной из наиболее важных задач конструкторского проектирования, возникающих на каждом из основных структурных иерархических уровней. Данная задача носит комбинаторно-логический характер и может выступать в качестве примера многокритериальной оптимизации. Особенностью многокритериальной оптимизации является наличие сразу нескольких критериев оценивания и сравнения эффективности возможных альтернатив.

Решение задачи компоновки блоков электронно-вычислительных средств основывается на их коммутационной схеме. Для алгоритмизации и формального решения задачи производится переход от коммутационной схемы к графу или гиперграфу [7]. В связи с данной постановкой задачи ее решение сводится к разбиению графа на заданное количество частей, что в свою очередь является NP-полной и NP-сложной задачей. В условиях современного уровня развития электронно-вычислительных средств использование стандартных алгоритмов компоновки является нецелесообразным в силу чрезмерно большого количества критериев оптимизации и количества элементов. Получение оптимального решения возможно только полным перебором различных вариантов разбиения.

Разбиение графа на части как правило может производиться несколькими методами. При использовании тривиального метода разбиения с использованием полного перебора находятся все возможные варианты разбиения, а затем из них отбирается наиболее подходящий.



Недостатком данного метода является то, что даже для современных ЭВМ он является чрезмерно ресурсозатратным (Рис. 1).



**Рис. 1 – Количество вариантов сочетаний.**

На рис. 1 показан график зависимости количества вариантов компоновки заданного количества элементов в 5 блоков. Количество вариантов компоновки находится по следующей формуле:

$$\tilde{C}_n^m = P_{m,n-1} = \frac{(n+m-1)!}{m!(n-1)!}.$$

При увеличении числа элементов или блоков электронно-вычислительного средства количество возможных вариантов их сочетания резко увеличивается и на их компоновку с использованием метода полного перебора потребуется предельно большое количество машинного времени. Таким образом, разработка нового метода компоновки блоков электронно-вычислительных средств, нетребовательного к машинным и временным ресурсам, является актуальной задачей.

Авторами предлагается генетический алгоритм компоновки блоков электронно-вычислительных средств, основанный на модифицированном методе генетического поиска.

Генетические алгоритмы входят в область мягких вычислений, которая включает в себя нечеткую логику, нейронные сети и так далее, в том числе и прочие эволюционные алгоритмы.

На рис. 2 представлена схема классического генетического алгоритма, используемого авторами для решения задачи компоновки блоков электронно-вычислительных средств.

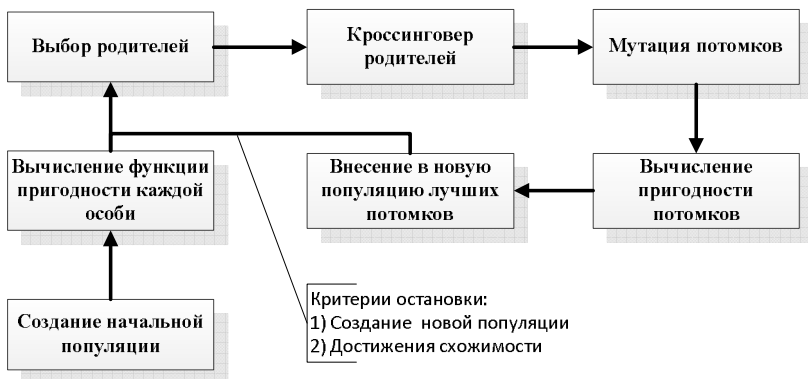


Рис. 2 – Схема генетического алгоритма.

Более подробного рассмотрения требуют наиболее важные характеристики алгоритма: функция полезности и структура хромосомы. Остальные характеристики генетического алгоритма подробно описываются в работах [1-6, 8-10].

На рис. 3 описана структура хромосомы, характеризующаяся зависимостью размера хромосомы от количества точек искомого пространства.

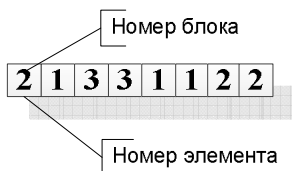


Рис. 3 – Структура хромосомы.

Функция полезности включает в себя элементы матрицы смежности и матрицы решений при выборе в качестве математической модели задачи компоновки взвешенный неориентированный граф. Она представлена ниже:

$$F = \frac{1}{2} \sum_{j=1}^r \left[ \sum_{i=1}^n \sum_{k=1}^n c_{ik} \xi_{ij} (1 - \xi_{ik}) \right] + \sum_{j=1}^r \sum_{i=1}^n c_{i0} \xi_{ij} ,$$

где  $\xi$  – матрица возможных решений,  $C$  – матрица смежности,  $r$  – число блоков,  $n$  – число элементов.

В настоящее время большинство промышленных САПР обходятся без компоновки блоков или накладывают на нее множество ограничений. Это вызвано высокой ресурсоемкостью и сложностью обработки необходимых массивов исходных данных. Разработанный авторами алгоритм компоновки блоков электронно-вычислительных средств позволяет эффективно учитывать множество входных ограничений, обладая при этом малой временной сложностью. Таким образом разработанный алгоритм является эффективным средством для решения задач компоновки блоков ЭВС большой размерности.\

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Аверченков, В.И. Эволюционное моделирование и его применение [Текст]: монография / В.И. Аверченков, П.В. Казаков. – Брянск: БГТУ, 2009. – 200 с.
2. Вороновский, Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности [Текст] / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Харьков, ОСНОВА, 1997. – 112 с.
3. Гладков, Л.А. Методы генетического поиска [Текст] / Л.А. Гладков, Л.А. Зинченко, В.В. Курейчик, В.М. Курейчик, Б.К. Лебедев, Е.В. Нужнов, С.Н. Сорокин. – Таганрог: Изд-во ТРТУ, 2002. – 122 с.
4. Гладков, Л.А. Генетические алгоритмы [Текст] / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2010. – 317 с.
5. Гладков, Л.А. Биоинспирированные методы в оптимизации [Текст]: монография / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М: Физматлит, 2009. – 384 с.
6. Емельянов, В.В. Теория и практика эволюционного моделирования [Текст] / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М: Физматлит, 2003. – 432 с.
7. Курейчик В.М. Математическое обеспечение конструкторско-го и технологического проектирования с применением САПР : учебник для вузов / В. М. Курейчик . – М. : Радио и связь, 1990 . – 352 с.
8. Курейчик, В.М. Поисковая адаптация: теория и практика [Текст] / В.М. Курейчик, Б.К. Лебедев, О.К. Лебедев. – М: Физматлит, 2006. – 272 с.
9. Панченко Т.В. Генетические алгоритмы [Текст]: Учебно-методическое пособие / под ред. Ю.Ю. Тарасевича. – Астрахань: АГУ, 2007. – 87 с.

10. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы [Текст]: Пер. с польск. И.Д. Рудинского / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия – Телеком, 2006. – 452 с.

**А.Н. САПРЫКИН, Е.Н. САПРЫКИНА**

Рязанский государственный радиотехнический университет

**ИССЛЕДОВАНИЕ ВЛИЯНИЯ БАЗОВЫХ ПАРАМЕТРОВ  
ГЕНЕТИЧЕСКОГО АЛГОРИТМА БАЛАНСИРОВКИ ПОТОКОВ  
ДАНЫХ В СЕТЯХ ДАТА-ЦЕНТРОВ НА СХОДИМОСТЬ**

*Рассматривается влияние размера популяции, числа особей и значения параметра мутации на сходимость генетического алгоритма, использующегося для балансировки потоков данных в сетях дата-центров.*

В настоящее время дата-центры для обслуживания каждого отдельного сервиса используют отдельные серверы или группы серверов. Это приводит к колебаниям спроса, что в свою очередь приводит к неэффективному использованию ресурсов сети. Как правило, дата-центры обеспечивают одновременное функционирование сервисов нескольких арендаторов, каждый из которых обладает различными требованиями. Разнообразие сервисов, делящих между собой ресурсы сети и вычислительные мощности, приводит к разнообразию моделей трафика в сетях дата-центров.

Одновременное использование каждым отдельным интернет-сервисом нескольких серверов и соединяющих их сетевых каналов сделало необходимым использование наиболее отказоустойчивой топологии сети дата-центров. Интернет-сервисы отличаются друг от друга по предъявляемым к сети требованиям, и каждый из них характеризуется особой моделью трафика. Большинство потоков невелики и недолговечны, в то время как существует небольшое количество крупных потоков, передающих наибольшее количество информации. Оба вида потоков обладают индивидуальными требованиями и поведением, что вызывает потерю пакетов и задержки запросов при их взаимодействии.

В работе [1] рассматривается возможность применения двухфазной многоканальной маршрутизации, предложенной Вэлиантом в 1982 году, в сетях дата-центров. Согласно данной концепции, маршрутизация в сети осуществляется через случайно выбранный промежуточ-

ный узел в маршруте до места назначения пакета. Такая рандомизированная маршрутизация называется двухфазной или двухканальной маршрутизацией.

Сети с двухфазной маршрутизацией обеспечивают высокую отказоустойчивость, что в сочетании с высокой производительностью делает их весьма перспективными для использования в сетях дата-центров. Такие сети позволяют поддерживать высокую пропускную способность сетевых каналов при передаче высокодинамичного трафика, а распределенная передача каждого потока по нескольким параллельным путям дает возможность равномерно сбалансировать нагрузку.

Для решения задачи балансировки нагрузки в сетях дата-центров авторами предлагается использовать модифицированный генетический алгоритм [4]. Его использование обусловлено неэффективностью применения стандартных методов поиска в силу большого количества комбинаций исходных данных.

Например, в сети с двухфазной маршрутизацией, состоящей из 32 узлов, передается 10 потоков. Потоки распределяются между двумя маршрутами передачи, при этом соотношение между частями потока определяется случайным образом. В этом случае число возможных комбинаций распределения потоков можно вычислить по формуле (1).

$$[(k-1)(25(k-2)+1)]^n, \quad (1)$$

где  $k$  обозначает число узлов в сети, а  $n$  – количество потоков. Тогда число неповторяющихся комбинаций распределения потоков составит  $4,68 \cdot 10^{43}$ .

Наиболее важными базовыми параметрами генетических алгоритмов являются размер популяции, число поколений и вероятность мутации. В большинстве случаев именно они определяют качество подобранных решений. Данный вопрос является одной из наиболее важных тем исследований эволюционных вычислений [5, 6]. На данный момент не существует однозначного мнения об оптимальных значениях базовых параметров.

Оптимальные значения базовых параметров генетического алгоритма зависят от характера и размерности конкретной решаемой задачи. В работах [2, 3] были предложены квазиоптимальные значения таких базовых параметров генетического алгоритма, как число полезных особей в популяции, конвергируемых поколений и вероятность мутации. Описанные решения могут быть применены только к хромосомам с ограниченной длиной на малых пространствах поиска. Найденные квазиоптимальные значения базовых параметров генетического алгоритма могут быть применены для решения задачи балансировки

нагрузки в сетях дата-центров, но только при условии наличия в сети малого числа узлов (не более 16).

В данной работе была рассмотрена проблема выбора оптимальных параметров генетического алгоритма для сетей дата-центров с числом узлов в сети в диапазоне от 16 до 32.

В результате проведенного исследования, были проведены симуляции нескольких тестовых сценариев. Эксперименты проводились на компьютере со следующими характеристиками: Intel Core i5 2,8 GHz, RAM 4Gb.

Эксперимент состоял из двух этапов. На первом этапе определялось оптимальное значение числа особей в популяции. На втором этапе на основе данных, полученных в результате экспериментов первого этапа, определялось оптимальное значение параметра вероятности мутации.

Первый этап. Используемые в экспериментах параметры генетического алгоритма были установлены следующим образом: размер популяции варьировался от 10 до 200 особей, вероятность скрещивания – 0,8, число поколений – 100, процент мутации – 5%. Вследствие стохастического характера генетических алгоритмов для каждого варианта было проведено 10 экспериментов.

В таблице 1 содержатся средние, лучшие и худшие результаты функции полезности для различных размеров популяции.

Эксперимент показал, что увеличение размера популяции с 10 до 100 особей в популяции значительно улучшает средние результаты функции полезности. Дальнейшее увеличение размера популяции не оказывает значительного влияния на точность полученных результатов.

**Таблица 1 – Значения функции полезности для различных значений размера популяции в сетях с двухфазной маршрутизацией из 16 и 32 узлов.**

| Размер популяции, особи | Функция полезности, условные единицы |                  |                  | Функция полезности, условные единицы |                  |                  |
|-------------------------|--------------------------------------|------------------|------------------|--------------------------------------|------------------|------------------|
|                         | Сеть из 16 узлов                     |                  |                  | Сеть из 32 узлов                     |                  |                  |
|                         | Средний результат                    | Худший результат | Лучший результат | Средний результат                    | Худший результат | Лучший результат |
| 10                      | 695                                  | 762              | 617              | 1621                                 | 1724             | 1458             |
| 20                      | 585                                  | 664              | 523              | 1523                                 | 1694             | 1281             |
| 30                      | 563                                  | 690              | 467              | 1449                                 | 1536             | 1359             |
| 40                      | 539                                  | 630              | 481              | 1377                                 | 1502             | 1249             |
| 50                      | 511                                  | 543              | 459              | 1409                                 | 1506             | 1318             |
| 60                      | 512                                  | 533              | 463              | 1350                                 | 1435             | 1275             |
| 70                      | 496                                  | 539              | 445              | 1389                                 | 1432             | 1313             |
| 80                      | 510                                  | 578              | 473              | 1363                                 | 1489             | 1203             |
| 90                      | 501                                  | 557              | 462              | 1358                                 | 1415             | 1297             |
| 100                     | 479                                  | 519              | 443              | 1363                                 | 1432             | 1243             |
| 110                     | 486                                  | 538              | 455              | 1364                                 | 1577             | 1291             |
| 120                     | 459                                  | 514              | 425              | 1425                                 | 1587             | 1317             |
| 130                     | 497                                  | 539              | 437              | 1444                                 | 1627             | 1371             |
| 140                     | 481                                  | 566              | 453              | 1405                                 | 1548             | 1249             |
| 150                     | 477                                  | 508              | 447              | 1378                                 | 1510             | 1269             |
| 160                     | 480                                  | 532              | 443              | 1405                                 | 1571             | 1272             |
| 170                     | 457                                  | 484              | 435              | 1389                                 | 1476             | 1314             |
| 180                     | 482                                  | 513              | 461              | 1379                                 | 1469             | 1288             |
| 190                     | 456                                  | 512              | 419              | 1364                                 | 1496             | 1311             |
| 200                     | 478                                  | 511              | 429              | 1352                                 | 1448             | 1229             |

Полученные числовые результаты были визуализированы (Рис. 1-2) для облегчения их восприятия. Согласно им, малый размер начальной популяции (10-20 особей) не позволяет генетическому алгоритму найти точное решение. Для достижения оптимального результата генетическому алгоритму требуется начальная популяция в размере не менее 40 особей.

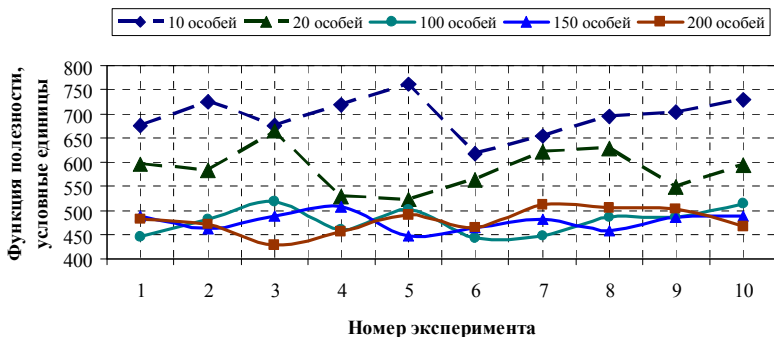


Рис. 1 – Средние значения функции полезности, полученные за 10 экспериментов для 10, 20, 100, 150 и 200 особей в популяции для сети с двухфазной маршрутизацией из 16 узлов.

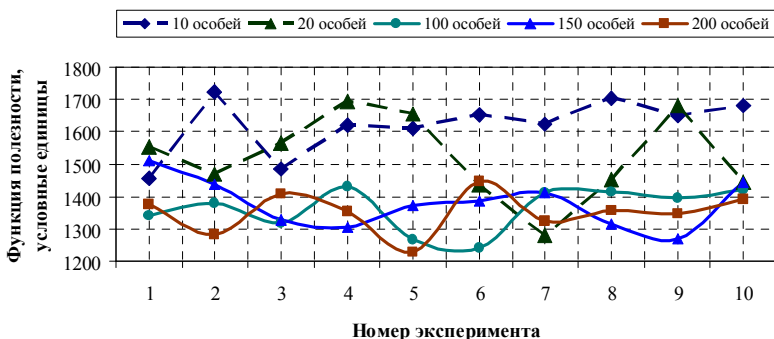


Рис. 2 – Средние значения функции полезности, полученные за 10 экспериментов для 10, 20, 100, 150 и 200 особей в популяции для сети с двухфазной маршрутизацией из 32 узлов.

Как видно из рис. 1-2, значения функции полезности, полученные в ходе экспериментов, в которых начальная популяция генетического алгоритма содержала менее 40 особей, отличались от значений, полученных в ходе остальных экспериментов, до 60%. Значения функции полезности, полученные в ходе экспериментов, в которых начальная популяция генетического алгоритма содержала от 100 до 200 особей, практически не подвергались отклонениям. Данные результаты экспериментов доказывают, что для нахождения глобального экстремума необходимо устанавливать значения базовых параметров генетического алгоритма на уровне определенного порогового значения.



Для выбора оптимального числа особей в популяции генетического алгоритма балансировки нагрузки в сетях дата-центров с числом узлов от 16 до 32 рекомендуется выставлять пороговое значение, равное 100 особям.

В данной работе не рассматривается варьирование параметра числа поколений генетического алгоритма. По умолчанию, данный параметр был выставлен, равным 100.

Второй этап. Используемые в экспериментах параметры генетического алгоритма были установлены следующим образом: вероятность мутации варьировалась от 1% до 50%, вероятность скрещивания – 0,8, число особей в популяции – 100, число поколений – 100. Вследствие стохастического характера генетических алгоритмов для каждого варианта было проведено 10 экспериментов.

**Таблица 2 – Значения функции полезности для различных значений вероятности мутации в сетях с двухфазной маршрутизацией из 16 и 32 узлов.**

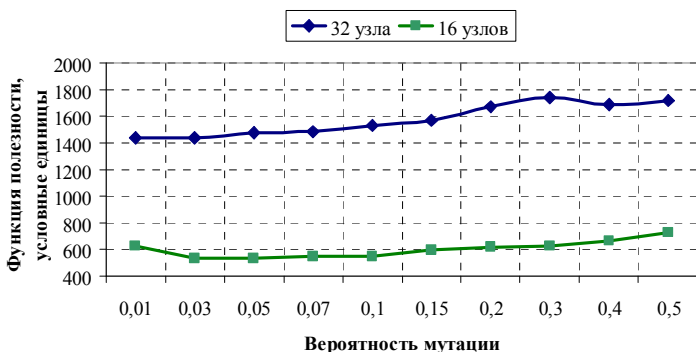
| Размер популяции, особи | Функция полезности, условные единицы |                  |                  | Функция полезности, условные единицы |                  |                  |
|-------------------------|--------------------------------------|------------------|------------------|--------------------------------------|------------------|------------------|
|                         | Сеть из 16 узлов                     |                  |                  | Сеть из 32 узлов                     |                  |                  |
|                         | Средний результат                    | Худший результат | Лучший результат | Средний результат                    | Худший результат | Лучший результат |
| 0,01                    | 695                                  | 762              | 617              | 1621                                 | 1724             | 1458             |
| 0,03                    | 585                                  | 664              | 523              | 1523                                 | 1694             | 1281             |
| 0,05                    | 563                                  | 690              | 467              | 1449                                 | 1536             | 1359             |
| 0,07                    | 539                                  | 630              | 481              | 1377                                 | 1502             | 1249             |
| 0,1                     | 511                                  | 543              | 459              | 1409                                 | 1506             | 1318             |
| 0,15                    | 512                                  | 533              | 463              | 1350                                 | 1435             | 1275             |
| 0,2                     | 496                                  | 539              | 445              | 1389                                 | 1432             | 1313             |
| 0,3                     | 626,9                                | 542              | 726              | 1435,8                               | 1323             | 1549             |
| 0,4                     | 535,5                                | 479              | 592              | 1438,9                               | 1278             | 1530             |
| 0,5                     | 534,2                                | 505              | 572              | 1477,4                               | 1372             | 1622             |

В таблице 2 содержатся средние, лучшие и худшие результаты функции полезности для различных значений вероятности мутации.

Эксперимент показал, что увеличение процента мутации выше 5-10% значительно ухудшает средние результаты функции полезности. Однако следует учитывать, что дальнейшее уменьшение процента мутации оказывает негативное влияние на значение функции полезности.

Полученные числовые результаты были визуализированы (Рис. 3) для облегчения их восприятия. Согласно им, малый размер начальной популяции (10-20 особей) не позволяет генетическому алгоритму найти точное решение. Для достижения оптимального результата ге-

нетическому алгоритму требуется начальная популяция в размере не менее 40 особей.



**Рис. 3** – Средние значения функции полезности, полученные за 10 экспериментов при проценте мутации от 1% до 50% для сетей с двухфазной маршрутизацией из 16 и 32 узлов.

Из рис. 3 видно, что при значениях вероятности мутации, выходящих за границы рекомендуемого диапазона 5-10%, результаты работы алгоритма значительно ухудшаются. Основываясь на полученных результатах можно сделать вывод о том, что грамотный выбор значения вероятности мутации генетического алгоритма балансировки нагрузки в сетях дата-центров значительно повышает его сходимость и быстродействие.

В заключение следует отметить, что размер популяции, число особей и значение параметра мутации оказывают большое влияние на сходимость генетического алгоритма, использующегося для балансировки потоков данных в сетях дата-центров. Установка базовых параметров генетического алгоритма в рамках определенных пороговых значений позволяет повысить сходимость алгоритма и сократить время поиска оптимального решения.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сапрыкина, Е.Н. Использование двухфазной многоканальной маршрутизации в сетях дата-центров / Е.Н. Сапрыкина, А.Н. Сапрыкин // Новые информационные технологии в научных исследованиях: материалы XXI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. – Рязань: РГРТУ, 2016. – С. 337–338.

2. Сапрыкин, А.Н. Нахождение оптимального числа полезных особей в популяции и конвергируемых поколений генетического алгоритма оптимизации простых многоэкстремальных функций: в 2-х т. / Сапрыкин А.Н., Акинина К.Д., Сапрыкина Е.Н. // Actualscience: сб. науч. тр. – Пенза, 2016. – Вып. 11. – С. 168-169. – 2 Т.

3. Сапрыкин, А.Н. Нахождение наилучших вероятностей мутации генетического алгоритма оптимизации простых многоэкстремальных функций : в 2-х т. / Сапрыкин А.Н., Акинина К.Д., Сапрыкина Е.Н. // Actualscience: сб. науч. тр. – Пенза, 2016. – Вып. 12. – 2 Т.

4. Сапрыкина, Е.Н. Программа балансировки нагрузки по схеме двухфазной маршрутизации в сетях дата-центров / Е.Н. Сапрыкина, А.Н. Сапрыкин // Новые информационные технологии в научных исследованиях: материалы XXI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. – Рязань: РГРТУ, 2016. – С. 341–343.

5. Diaz-Gomez P.A., Hougen D.F. Initial Population for Genetic Algorithms: A Metric Approacs // International Conference on Genetic and Evolutionary Methods, GEM 2007. P. 43–49.

6. Piszcz A., Soule T. Genetic programming: Optimal population sizes for varying complexity problems // Genetic and Evolutionary Computation Conference, 2006. P. 953–954.

**А.Н. САПРЫКИН, Е.Н. САПРЫКИНА**

Рязанский государственный радиотехнический университет

**МЕТОДЫ И АЛГОРИТМЫ БАЛАНСИРОВКИ ПОТОКОВ  
ДАНЫХ В СЕТЯХ ДАТА-ЦЕНТРОВ**

*Рассматриваются основные методы и алгоритмы балансировки потоков данных в сетях дата-центров, их достоинства и недостатки, а также проводится их сравнительный анализ с точки зрения отказоустойчивости.*

Возникшая парадигма облачных вычислений привела к созданию центров обработки данных, которые состоят из сотен тысяч серверов и которые способны поддерживать большое количество различных услуг. Поскольку затраты на развертывание и обслуживание центра обработки данных чрезвычайно велики [1], немаловажным является достижение высокого коэффициента производительности путем балансировки потоков данных.

Основными задачами балансировки являются распределение нагрузки между серверами и повышение их отказоустойчивости.

Балансировка может осуществляться на разных уровнях: канальном, сетевом или транспортном; у каждого метода балансировки есть как достоинства, так и недостатки.

Среди достоинств балансировки на канальном уровне следует отметить, что она является независимой от протоколов вышестоящих уровней модели OSI. Однако, при данном методе балансировки существует ограничение на пропускную способность канала, отводимую под входящий трафик. Данный недостаток обусловлен необходимостью размещения серверов внутри одного сегмента сети.

Балансировка на сетевом уровне также не зависит от протоколов вышестоящих уровней модели OSI, но имеет другой недостаток: обратный трафик повышает нагрузку на канал. Особенностью балансировки нагрузки на сетевом уровне является наличие одного IP-адреса сервера, который поддерживается несколькими физическими серверами.

Балансировка, осуществляемая на транспортном уровне, похожа на балансировку на сетевом уровне. Она отличается лишь тем, что учитывает в своей работе порты источника и назначения. Кроме того, использование балансировки на транспортном уровне позволяет избавиться от необходимости приобретения специального дополнительного оборудования. С другой стороны, в этом случае нельзя избежать установки дополнительного программного обеспечения (например, роутера BGP).

На данный момент существует множество алгоритмов балансировки потоков данных. В данной работе рассматриваются основные из них.

Одним из распространенных алгоритмов балансировки потоков данных является алгоритм кругового обслуживания или Round Robin. Его суть состоит в том, что он осуществляет круговой перебор запросов. Они передаются от одного сервера к другому до достижения последнего, после чего цикл начинается сначала.

В числе преимуществ этого алгоритма стоит отметить его независимость от протоколов высокого уровня и текущей нагрузки на сервера. Однако, он обладает и некоторыми недостатками. Так как алгоритм Round Robin не учитывает текущую нагрузку каждого отдельно взятого сервера, данное достоинство превращается в недостаток в ситуации, когда нагрузка на одни узлы сети является предельно допустимой, в то время как остальные узлы простаивают или загружены лишь на малую долю своей обрабатывающей способности. Данный недостаток алгоритма Round Robin устраняется его более совершенной версией Weighted Round Robin.

Принцип работы алгоритма Weighted Round Robin аналогичен его исходному алгоритму, однако, во избежание равномерной нагрузки, каждому серверу присваивается его весовой коэффициент, отражающий степень его загруженности и круговая обработка запросов осуществляется в соответствии с этими коэффициентами, отправляя потоки данных на менее загруженные сервера.

К возможным недостаткам данных алгоритмов следует отнести также то, что они обладают низкой отказоустойчивостью и не учитывают число активных подключений.

Аналогичным подходом к балансировке обладает алгоритм Least Connections. Так же, как и предыдущий алгоритм, он присваивает серверам весовые коэффициенты, но помимо них учитывает также и количество подключений к серверам.

Существует также случайный алгоритм Random. Как следует из названия, этот алгоритм использует генератор случайных чисел для выбора наиболее эффективных серверов-адресатов.

Другим способом решения задачи балансировки является использование метрик. Его реализует несколько различных алгоритмов, например, алгоритм Spillover. Он выбирает для потока маршрут с самым низким значением метрики. От значения метрики зависит порядок, в котором сетевые потоки будут перенаправлены на другие каналы в случае превышения пропускной способности текущего канала.

Особенностью алгоритма Spillover является создание им сразу нескольких возможных маршрутов для потоков данных в сети. Эти альтернативные маршруты ранжируются в зависимости от значения метрики; предпочтение отдается маршрутам с наименьшим значением. Настройки алгоритма позволяют ввести в них пороговую величину Spillover Setting, которая определяет, не превышает ли заданный порог на текущем маршруте, и в случае его превышения отправляет поток данных на следующий альтернативный маршрут.

В качестве недостатка алгоритма Spillover необходимо упомянуть невозможность указания одинаковых значений метрик для нескольких маршрутов.

Существующие методы и алгоритмы балансировки потоков в сетях дата-центров не полностью справляются со своей задачей вследствие того, что эффективность их работы напрямую зависит от частоты обновления информации о состоянии сети. Если алгоритм в качестве исходных данных использует устаревшую информацию, его эффективность и быстродействие значительно снижаются. Каждый раз, при принятии нового решения информация о состоянии сети нуждается в

обновлении, что может привести к значительной потере пропускной способности всей сети из-за обмена служебными сообщениями [2].

Частота обновления информации о состоянии сети в большинстве случаев ограничивается во избежание чрезмерно высокой загрузки сети. Однако, слишком малое значение данного параметра может привести к ситуации, когда алгоритм делает вывод, основываясь на информации, последнего обновления, которое происходило слишком давно. Тогда маршрут, отбираемый алгоритмом как предпочтительный, теоретически мог за время, прошедшее с последнего обновления, стать наименее эффективным. Таким образом, корректный выбор значения параметра частоты обновления информации о состоянии сети является одним из обязательных условий эффективной балансировки потоков в сетях дата-центров.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. A. Greenberg, J. Hamilton, D. A. Maltz, P. Patel, .The Cost of a Cloud: Research Problems in Data Center Networks,. ACM SIGCOMM Computer Communication Review, 39(1):68-73, Jan. 2009.

2. P. Van Mieghem, F.A. Kuipers, T. Korkmaz, et al. Qaulity of Service routing // Lecture Notes in Computer Science. – 2003. – Volume 2856. – 80 – 117 pp.

### **А. О. САПРЫКИНА, А. Н. САПРЫКИН**

Рязанский государственный университет имени С.А. Есенина  
Рязанский государственный радиотехнический университет

### **ОСНОВНЫЕ ТИПЫ ЭЛЕКТРОННОГО ПОРТФОЛИО**

*Рассматриваются четыре основных типа электронных портфолио, их особенности и сложности их внедрения в учебный процесс.*

Технология электронного портфолио, представляющего собой оцифрованное собрание образовательных материалов и сертификатов, нашла свое место в современной системе образования. Она используется во многих странах мира, включая и Россию. Такому распространению данной технологии способствовало наличие у нее множества достоинств, позволяющих значительно повысить качество обучения и оценивания образовательных достижений.

Электронное портфолио является не столько простым собранием разрозненных пользовательских файлов, сколько административным

инструментом структурирования учебных работ, создающихся разными способами и в разных программах, и инструментом хранения и демонстрации свидетельств достижения того или иного образовательного уровня путем отображения различных сертификатов.

Особенности каждого конкретного электронного портфолио зависят от целей его создания. Можно выделить шесть основных целей, достижению которых способствует данная технология:

1. планирование учебного плана;
2. приобретение знаний, умений и навыков;
3. отслеживание процесса продвижения по конкретному курсу;
4. поиск работы;
5. оценивание курса;
6. динамический контроль и оценивание успеваемости.

Существует множество моделей электронного портфолио, разными способами комбинирующих направленность на одну или сразу несколько из перечисленных целей. Их можно разделить на четыре большие группы: **портфолио обучающегося**, **портфолио преподавателя**, **портфолио учебного заведения** и **пожизненное портфолио**.

**Портфолио обучающегося** зародилось на основе бумажных портфолио студентов, занимающихся искусством или связанных с большими объемами рукописного и печатного текста [3]. Сложенные в папку (а точнее, портфолио) работы обучающихся позволяли им эффективно демонстрировать свои достижения и рефлексировать на тему собственного прогресса. Электронное портфолио обучающегося реализует те же функции, перенося их в электронную форму и расширяя доступные ранее возможности. Иными словами, электронное портфолио обучающегося призвано не только хранить и демонстрировать работы обучающегося, но и служить базой для его личностного и профессионального роста путем саморефлексии и самооценивания.

**Портфолио преподавателя** происходит от учебных планов и курсовых материалов. Так же, как и портфолио обучающегося, оно может выступать в роли свидетельства овладения определенными знаниями, умениями и навыками, а также демонстрировать историю профессионального роста преподавателя [1]. Данный вид портфолио может использоваться как для демонстрации личных разработок коллегам, так и студентам или потенциальным работодателям. Портфолио преподавателя как правило содержит основные учебные материалы, разработанные автором, его учебные планы, лекции, практические задания и проч., а также свидетельства о повышении квалификации или сертификаты соответствия занимаемой должности.

**Портфолио учебного заведения** объединяет в себе отдельные портфолио обучающихся и преподавателей в большую базу данных [2]. Помимо двух перечисленных видов портфолио, в него также могут быть включены и портфолио отдельных подразделений и кафедр учебного заведения. Целью создания таких портфолио является объединение всех материалов и данных учебного заведения в единую систему, позволяющую значительно повышать возможности корректировки, рефлексии и самообучения у ее пользователей.

**Пожизненное портфолио**, как следует из его названия, содержит в себе всю учебную, профессиональную и личную информацию своего пользователя. Данная информация аккумулируется на протяжении всей жизни и позволяет проследить не только успешность обучения и динамику развития пользователя, но и процесс формирования и корректировки личной образовательной и профессиональной траектории.

При всей эффективности электронных портфолио как образовательных инструментов, они обладают и рядом недостатков. Рассмотрим основные сложности, связанные с внедрением в учебный процесс электронных портфолио разных типов.

При использовании в учебном заведении портфолио обучающегося возникает целый вопросов, связанных с особенностями его использования. Руководство учебного заведения следует четко определить, будет ли электронное портфолио являться основным способом ведения учебной отчетности обучающегося и будет ли оно являться свидетельством его профессиональной компетентности по окончании обучения. Данный вопрос возникает из-за того, что электронное портфолио с одной стороны содержит подтверждение сформированности у обучающегося определенных навыков, но с другой стороны обучаемый всегда может его изменить или дополнить, и в таком случае появляется необходимость в их периодической аккредитации. Помимо этого, хранение электронного портфолио требует места на серверах учебного заведения, и так как со временем его объем только увеличивается, необходимо определить, как долго электронные портфолио обучающихся должны храниться в учебном заведении после их выпуска. Также следует отметить вопросы об авторских правах на содержимое портфолио и его объективного оценивания.

Основной проблемой при создании портфолио преподавателя является его структуризация. Как правило, преподаватели обладают большим количеством разнородной образовательной информации, которая при отсутствии должной организации может легко запутать обучающихся и преподавателей, использующих данное портфолио в



качестве учебного средства. Также в случае использования преподавателями не только своих личных разработок, вновь встает вопрос об авторских правах.

Портфолио учебного заведения обязательно требуют создания сложной системы прав доступа к различным работам или даже целым портфолио для разных участников учебного процесса. В силу своей масштабности и большого объема информации, являющегося гораздо большим, чем при использовании электронных портфолио обучающихся и преподавателей, электронное портфолио учебного заведения предъявляет еще большие требования к организации и хранению данных. В этом случае, как правило, требуется особый сотрудник, занимающийся обеспечением работы всей системы электронных портфолио.

Пожизненное электронное портфолио предъявляет требования, схожие со всеми выше перечисленными в силу своей протяженности, большого объема данных, необходимости в аккредитации и постепенного изменения целей владельца портфолио в течении жизни.

Электронные портфолио разных типов обладают множеством сильных сторон, которые эффективно реализуются в зависимости от целей, но для эффективного внедрения данной технологии в учебный процесс следует также учитывать и их особенности и основные проблемы, возникающие в ходе их использования.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Kahn, S. Linking Learning, Improvement and Accountability: An Introduction to Electronic Institutional Portfolios // *Electronic Portfolios: Emerging Practices in Student, Faculty, and Institutional Learning*, Cambridge, B. L. et al., eds. Washington, D.C.: AAHE, 2001, pp. 135–158.

2. Thomson, B. UBC Teaching e-portfolio/FAQ // UBC Wiki [Электронный ресурс], URL:[http://careo.elearning.ubc.ca/cgi-bin/wiki.pl?UBC\\_Teaching\\_EPortfolio/FAQ](http://careo.elearning.ubc.ca/cgi-bin/wiki.pl?UBC_Teaching_EPortfolio/FAQ).

3. Yancey, K. B. “Digitized Student Portfolios // *Electronic Portfolios: Emerging Practices in Student, Faculty, and Institutional Learning*. Cambridge, B. L. et al., eds. Washington, D.C.: AAHE, 2001, pp. 15–30.

А.Л. САФОНОВ

Рязанский государственный радиотехнический университет

## ВЗАИМОДЕЙСТВИЕ СЕТЕВЫХ УСТРОЙСТВ ДРУГ С ДРУГОМ В ГЕТЕРОГЕННЫХ СЕТЯХ. CWMP

*Рассматриваются имеющиеся протоколы автоматизации управления устройствами в корпоративных/провайдерских сетях, анализируются некоторые основные положения из протокола CWMP. Исследуются механизмы, обеспечивающие связь управления между разнородными устройствами.*

Производство сетевого оборудования не является монополией — все больше и больше компаний выпускают на рынок сетевых коммуникаций различные устройства, часто несовместимые между собой в плане настройки. При этом большинство крупных производителей пытаются продвинуть собственную модель управления устройством в виде негласного стандарта. На этом поле имеет определенный успех компания Cisco, которая смогла в большинстве случаев обеспечить общий подход к настройке своих устройств.

Эта проблема является актуальной, так как при администрировании больших сетей накладные расходы по обслуживанию и настройке оборудования сильно возрастают. В особо запущенных случаях приходится держать целый штат сетевых администраторов, которые и занимаются ручной настройкой.

Обычно все корпоративные устройства (коммутаторы, брандамауэры, ...) поддерживают автоматизированную настройку (либо единовременное обновление конфигурации, либо возможность гибкой настройки через консольное управление или скрипты). Основную проблему представляют SOHO-устройства (конечные устройства) — в них редко предусматривается возможность автоматизированного управления.

Для этого был придуман протокол CWMP — *CPE WAN Management Protocol* — протокол централизованного управления абонентским оборудованием. Под *CPE* понимается *Customer Premises Equipment* — оборудование, установленное в помещении абонента/клиента. *WAN (wide area network)* — сеть, охватывающая большие территории.

### CWMP

Протокол CWMP был создан специально для удаленного управления конечным оборудованием через глобальную сеть. Одной из спецификаций этого протокола является *TR069*. Он разработан некоммерческой организацией *Broadband Forum*, которая составляет техниче-

ские отчеты (*TR* - technical report) для сетевых инфраструктур.

Основной особенностью *TR069* является возможность управления большим числом разнородных устройств, соединенных глобальной сетью Интернет.

Спецификация *TR069* вводит несколько определений:

- *CPE* — конечное клиентское оборудование;
- *ACS* (*auto-configuration server*) — сервер автоконфигурирования.

Каждое *CPE* регистрируется на *ACS*, при этом между сервером и клиентом организуется канал связи, по которому они могут обмениваться различной информацией. К примеру, сервер *ACS* в состоянии:

- производить начальную настройку и вносить изменения в настройки уже работающего устройства;
- удаленно обновлять ПО/прошивку;
- просматривать статистику устройства, журналы логирования;
- диагностировать неисправности устройства;
- исполнять хранимые процедуры *CPE*.

Спецификация *TR069* определяет механизмы взаимодействия устройств. Особое внимание уделено спецификации вызовов процедур (*RPC*). Для каждого типа устройства (*CPE*, *ACS*) описываются определенные методы, которые тот должен или может поддерживать. Например, методы *CPE*:

- *GetRPCMethods* — список поддерживаемых методов;
- *GetParameterNames* — получение общей информации о доступных параметрах конфигурации;
- *GetParameterAttributes* — получение свойств каждого параметра (напр. права чтения/записи, тип поля, границы значений и т.п.)
- *GetParameterValues* — чтение текущих значений различных параметров от *CPE*;
- *SetParameterValues* — запрос на применение новых параметров *CPE*;
- *SetParameterAttributes* — запрос на изменение свойств параметров;
- *AddObject* — добавление нового объекта в дереве конфигурации;
- *DelObject* — удаление существующего объекта из

дерева;

- *Download* — запрос на загрузку (и применение) нового ПО или файла с конфигурацией;
- *Reboot* — запрос на перезагрузку *CPE*.

Каждый метод подробно описан, определены все параметры каждого метода и ожидаемое поведение в различных ситуациях.

### **Device Datamodel**

Для обмена параметрами конфигурации выработан отдельный отчет *TR181*, определяющий требуемые свойства построения дерева конфигурации устройств - *датамодели*.

Основополагающим понятием является то, что все доступные настройки логически группируются в виде дерева. Корнем дерева является поле *Device*. От корня расходятся множество ветвей, соответствующих конкретным аспектам настройки. Например, секция *Device.WiFi* отвечает за управление WiFi-радиопередатчиками и параметрами WLAN-сетей.

*Датамодель* позволяет описывать настройки и для динамических сущностей (такие на языке *TR181* именовываются *инстансами*). Например, настройки MAC-фильтрации или правил маршрутизации определяются в виде *инстансов*, так как имеют одинаковые опции и могут быть динамически добавлены или удалены. Для таких объектов применимы методы *AddObject* и *DeleteObject*. При создании любого объекта ему присваивается уникальный *номер инстанса*, по которому этот объект и закрепляется в дереве настройки. Пример: некоторый WiFi-радиопередатчик описывается в виде *инстанса* и имеет абсолютный путь *Device.WiFi.Radio.1*. При удалении некоторого *инстанса* использованный им номер больше не может быть использован.

Организацией *Broadband Forum* предоставляется в свободном доступе стандартная *датамодель* устройства, описывающая большинство базовых параметров сетевого оборудования. При этом спецификация *TR181* определяет и то, как можно добавить специализированные параметры настройки, зависящие от производителя устройства. В общем виде имя такого поля будет выглядеть как *X\_<OUI>\_<имя>*, где *OUI (organization unique identifier)* — 6 шестнадцатиричных цифр, обозначающих организацию.

Так как каждое *CPE* должно поддерживать методы *GetParameterNames* и *GetParameterAttributes*, то любой *ACS* в состоянии обработать любой параметр настройки устройства, в том числе и специализированный для конкретного производителя. Разношерстность устройств теперь определяется только наполнением дерева кон-

фигурации, механизмы управления остаются общими.

Таким образом решается проблема управления большим числом устройств. Большинство операторов и провайдеров устанавливают свои собственные ACS-сервера, которые занимаются мониторингом и управлением массой устройств, снижая накладные расходы на поддержку каждой отдельной модели отдельно.

В настоящее время большое число производителей заявляют поддержку спецификации TR069 в той или иной степени. Можно заметить, как некоторые компании, ранее имевшие собственный формат конфигурации, сейчас предоставляют встроенные TR-клиенты, являющиеся прослойкой, преобразующей внутреннее представление в стандарт *датамодели*. В некоторых случаях внутреннее представление заменяется на формат, подобный или соответствующий спецификации TR181, что упрощает поддержку устройств, перенос конфигураций между различными моделями и значительно облегчает TR069-совместимую прослойку. Можно предположить, что в недалеком будущем SWMP станет базовым требованием при выходе новой продукции на рынок.

### Заключение

В данной статье был описан протокол SWMP, некоторые спецификации, связанные с автоматизированной настройкой и обменом данными, рассмотрено понятие *датамодели* устройства. Такая модель позволяет не привязываться к производителю при управлении устройством. Общий каркас и методы доступа позволяют получать представление о возможностях устройств, имея на руках только дерево конфигурации. Это может пойти на пользу как крупным организациям, упрощающим и унифицирующим внутренний формат данных, так и небольшим компаниям, выходящим на рынок с новым устройством, но привычным методом управления.

Наступил тот момент, когда большинство производителей и потребителей телекоммуникационного оборудования, наконец, собрались вместе в рамках группы *Broadband Forum* и пришли к общему решению проблемы разного формата данных различных устройств.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. TR-069 Amendment [Электронный ресурс]. — Режим доступа: [https://www.broadband-forum.org/technical/download/TR-069\\_Amendment-2.pdf](https://www.broadband-forum.org/technical/download/TR-069_Amendment-2.pdf) — (Дата обращения: 18.04.2017).
2. TR-181 Amendment [Электронный ресурс]. — Режим доступа: [https://www.broadband-forum.org/technical/download/TR-181\\_Issue-2\\_Amendment-2.pdf](https://www.broadband-forum.org/technical/download/TR-181_Issue-2_Amendment-2.pdf) — (Дата обращения: 18.04.2017).

3. Cisco TR069-client [Электронный ресурс]. — Режим доступа: [http://www.cisco.com/c/en/us/td/docs/ios/bbds1/configuration/guide/bba\\_tr069\\_agent.html](http://www.cisco.com/c/en/us/td/docs/ios/bbds1/configuration/guide/bba_tr069_agent.html) — (Дата обращения: 18.04.2017).

4. Mikrotik TR069-client [Электронный ресурс]. — Режим доступа: <https://wiki.mikrotik.com/wiki/Manual:TR069-client> — (Дата обращения: 18.04.2017).

### **О.И. СКУРАТОВ, П.Э. СКУРАТОВА**

Рязанский государственный радиотехнический университет

### **ГИБРИДНАЯ ВЕТРОСОЛНЕЧНАЯ СИСТЕМА**

*Рассматривается возможность соединения ветряной и солнечной энергетических установок.*

В настоящее время по подсчетам Министерства энергетики на 70% территории России отсутствует централизованное электроснабжение. Причиной являются большая удаленность населенных пунктов друг от друга, из-за чего подключение поселений к энергосистемам весьма дорогостояще. В данных районах проживает большое количество человек, которые пользуются в основном топливными генераторами. Стоимость вырабатываемой топливными установками электроэнергии довольно высокая, что связано в основном с трудоёмкостью доставки топлива. Это говорит о необходимости использования в таких областях систем, не требующих подключения к сети электропитания. Примерный состав подобных систем: генератор энергии, контроллер (ШИМ, МРРТ), реле, инвертор.

Ветряные электростанции вырабатывают электричество за счет энергии перемещения ветра. Различают два типа ветряков: с горизонтальной или вертикальной осями вращения. Ветряная электростанция с горизонтальной осью вращения - это мачта, наверху которой размещается корпус с редуктором и генератором. К оси редуктора ветряка прикрепляются лопасти. Корпус электростанции поворачивается в зависимости от направления ветра. Ветряные электростанции с вертикальной осью вращения менее популярны. Сам генератор устанавливается под мачтой, и одна из главных особенностей, отсутствует необходимость ориентации электростанции на ветер. Основными преимуществами ветряных электростанций являются неисчерпаемый источник энергии ветра - природа и тот факт, что ветряные электростанции не загрязняют окружающую среду. Основным недостатком рассматри-

ваемого источника энергии можно отнести проблему нестабильной природы ветра, техническое решение которой, является главной задачей при создании ветряных электростанций.

В сравнении с ветряными электростанциями у солнечных более сложный процесс изготовления, в связи с чем их стоимость будет выше. У таких систем есть преимущества: работают бесшумно и без перерыва, более долговечны, солнечный свет более постоянный, чем порывы ветра. К основному недостатку рассматриваемого источника энергии можно отнести продолжительность дня в зависимости от часового пояса. Объединив ветрогенераторы и солнечные батареи, мы получим систему, в которой будут компенсированы недостатки получения энергии. В таких системах одним из важнейших составляющих являются контроллеры заряда. Они служат для контроля и регулирования заряда аккумуляторных батарей. Полное разряжение, как и чрезмерная зарядка, влияют на дальнейшую работу аккумуляторных батарей. Для предохранения батарей от этих нагрузок и служит контроллер. При максимальной зарядке аккумуляторной батареи с помощью контроллеров уровень тока будет понижен, при понижении заряда до критических значений подача энергии будет остановлена.

Существуют несколько типов регуляторов: On/Off, ШИМ и МРРТ.

Контроллеры On/Off являются простыми и недорогими. Когда заряд аккумулятора достигает предельного значения, контроллер разрывает соединение между солнечной панелью и батареей с помощью реле. В действительности батарея не полностью заряжена, что оказывает влияние на дальнейшую работоспособность аккумулятора. Поэтому, несмотря на низкую стоимость, лучше не использовать регулятор данного типа. Для ШИМ (PWM) контроллеров применена технология широтно-импульсной модуляции. Преимуществом является прекращение заряда аккумуляторной батареи без отсоединения солнечных модулей, что позволяет продолжить зарядку аккумуляторной батареи до максимального уровня. МРРТ (Maximumpowerpointtracker) контроллер считается самым эффективным. Он отслеживает максимальный энергетический пик и понижает напряжение, но увеличивает силу тока, не изменяя мощность. Благодаря высокому коэффициенту полезного действия МРРТ – контроллеры сокращают срок окупаемости солнечных станций.

Гибридным контроллером считается контроллер, использующий энергию ветра и солнца. Его преимуществом является возможность использование двух источников тока совместно или попеременно. Возможно, подобрать контроллер с нужными характеристиками для

каждого потребителя. Существуют контроллеры с возможностью подключения к компьютеру. Одним из оптимальных вариантов платформы контроллера служит платформа фирмы Arduino. Плюсов достаточно много. Основным преимуществом является доступность, ведь программная оболочка бесплатна. Благодаря открытой архитектуре системы проблем с дополнением линейки не возникнет. Данные контроллеры поддерживают двигатели с напряжением до 12 вольт, можно подключить реле.

Инвертор, это устройство, преобразовывающее постоянный ток в переменный. При этом частота и/или напряжение меняется. Схем инверторов достаточно большое количество, самыми часто встречающимися являются три типа. Первый тип – это мостовые инверторы без трансформатора, применяются для установок с высокими напряжениями (от 220 до 360 вольт). Ко второму типу относят инверторы с нулевым выводом трансформатора, используют в системах с низким напряжением (12–24 вольт). И третьим типом являются мостовые инверторы с трансформатором. Их применяют для обширных диапазонов напряжений мощности (48 вольт).

Таким образом, для обеспечения бесперебойного автономного электроснабжения объекта, уменьшения необходимой мощности ветрогенератора и солнечной батареи, емкости аккумуляторной батареи, улучшения режимов работы станции, во многих случаях целесообразно использование гибридной ветросолнечной электростанции. Наиболее заметны преимущества гибридных станций при круглогодичном их использовании. При этом в зимнее время наибольшая выработка электроэнергии приходится на ветроэлектрическую установку, а летом - на солнечные батареи. В качестве резервного источника электроснабжения в ветросолнечную систему может быть добавлен дизель- или бензогенератор. Тогда состав гибридной системы в общем случае будет следующим: ветросолнечная станция; резервный бензо- или дизельгенератор; блок бесперебойного питания со встроенным контроллером заряда-разряда аккумуляторной батареи; аккумуляторная батарея.

### **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Электронный ресурс <http://solntsepek.ru/kontrollery-zaryada/kontroller-zaryada-solnechnyx-batarej.html>.
2. Электронный ресурс - <http://rg.ru/2011/09/30/energo.html>



**А.И. СОБОЛЕВА, А.Ю. ГРОМОВ**

Рязанский государственный радиотехнический университет

## **ИСПОЛЬЗОВАНИЕ OLAP-ТЕХНОЛОГИЙ В СИСТЕМАХ МОНИТОРИНГА ОКРУЖАЮЩЕЙ СРЕДЫ**

*В статье рассматриваются технологии построения многомерных кубов при проектировании информационных систем с использованием компонентов Business Intelligence Development Studio.*

Основными функциями системы комплексного мониторинга окружающей среды являются организации сбора, передачи, обработки и хранения информации о состоянии окружающей среды и природных ресурсов, а также проведение анализа состояния окружающей среды на основе выполненных наблюдений. Задача анализа может быть решена методом извлечения и визуализации данных. Аналитик формулирует запрос к информационной системе, извлекает нужную информацию из источников и просматривает получившиеся данные, на основе которых делаются выводы – результаты анализа. OLAP – один из способов визуализации данных [1].

OLAP предоставляет средства для быстрого и удобного доступа, просмотра и анализа больших объемов информации. Работа в системах поддержки многомерного оперативного анализа ведется с естественно организованной многомерной моделью данных, представленных в виде OLAP-кубов. Эта структура данных обеспечивает быструю работу с запросами с большими объемами агрегации, посредством предварительного формирования агрегатов. Эта особенность устраняет недостаток реляционных баз данных (неэффективность при агрегировании и поиске при работе с большими объемами данных), что делает OLAP-системы актуальными при анализе данных.

Цель данной работы – практическое изучение возможностей осмысленного управления и анализа данных посредством разработки OLAP-куба и использование его в ИС.

Разработка куба в среде Business Intelligence Development Studio включает следующие этапы [2]:

1. Подготовку базы данных Analysis Services и создание представления источника данных.

2. Определение измерений в базе данных Analysis Services, а именно основных атрибутов анализируемого бизнес-процесса, которые используются для хранения информации о бизнес-сущностях, а также добавление иерархий и связей атрибутов измерения.

3. Добавление мер – зависящих от измерений агрегатных функций.
4. Создание куба через Мастер куба.
5. Добавление вычислений при помощи языка многомерных выражений MDX.
6. Развертывание базы данных SQL Server Analysis Services из Business Intelligence Development Studio.

На рис. 1 представлена структура куба с мерами таблицы проводимых исследований и измерениями - объекты наблюдения, сотрудники, санитарно-гигиенические показатели, дата исследования. Схема хранилища – «снежинка».

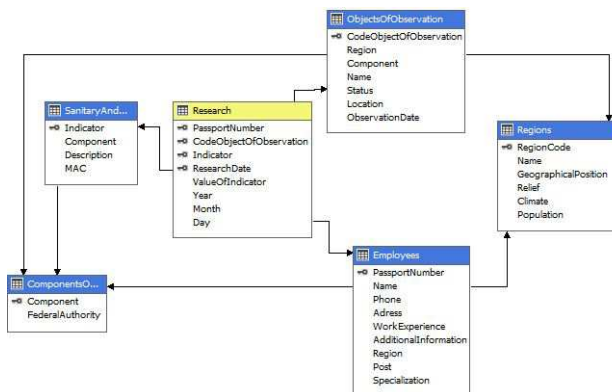


Рис. 1 – Структура куба.

Наиболее существенными критериями при построении кубов являются режим хранения данных и уровень агрегирования. Для разрабатываемого куба использовались стандартные настройки режима хранения – MOLAP (данные-факты и агрегаты хранятся в оптимизированном многомерном формате на OLAP-сервере), при котором достигается максимальная скорость обработки запросов. Создание группы агрегатов (предварительно рассчитанных сумм данных таблицы фактов для определенных комбинаций уровней из каждого измерения) было проведено с помощью мастера Aggregation Design. Разработка может происходить, пока не будет достигнут необходимый прирост производительности в оптимизации запросов или указанный объем создаваемых агрегатов.

Работа с OLAP-системами может быть построена на основе двух схем:

1. OLAP-средства встроены в настольные приложения.

2. Применение двухступенчатой схемы "клиент-сервер". Сервер обеспечивает непосредственно извлечение информации из СУБД. Специализированное приложение-клиент предназначено для удобного и эффективного просмотра кубов и выявления аналитических закономерностей. В линейке продуктов Microsoft серверная часть представлена в лице Microsoft Analysis Services, которые входят в MS SQL Server.

На рис. 2 показано использование OLAP-куба в разрабатываемой информационной системе мониторинга. Связь со службами Analysis Services ведется через библиотеки ADOMD.NET, работа с кубом – через PivotGrid, MdxDataSelector вендора Infragistics [3].

Мониторинг состояния окружающей среды (аналитик)

Соединение | Справочная информация

OLAP | Отчетность

Filters

Research Count | Component

Year-Month

|      |    | водный объект | воздух | All |
|------|----|---------------|--------|-----|
| 2011 | 11 | 1             | 1      |     |
|      |    |               |        | 1   |
| 2012 | 3  | 1             | 1      |     |
|      | 4  |               |        | 1   |
| 2012 | 5  | 2             | 2      |     |
|      |    |               |        | 4   |
| 2013 | 9  | 5             | 14     |     |
| 2015 | 10 | 11            | 21     |     |
| 2016 | 9  | 10            | 19     |     |
| 2017 | 2  | 7             | 9      |     |
| All  | 30 | 38            | 68     |     |

SSAS\_OLAPCube\_EcologicalMonitoring

Ecological Monitoring

- Measures
  - Research
    - Maximum Value Of Indicator
    - Minimum Value Of Indicator
    - Research Count
- Employees
- Objects Of Observation
  - Code Object Of Observation
  - Component
  - Name
  - Regions - Name
  - Status
- Research
  - Sanitary And Hygienic Indicators

Рис. 2 – Использование куба в информационной системе.

OLAP предоставляет пользователю агрегатные данные для различных выборок из исходного набора в удобном для восприятия и анализа виде, а формирование запросов путем перетаскивания мер и измерений позволяет быстро получить необходимую информацию.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бергер, А.Б. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных / А.Б. Бергер, И. Горбач. – БХВ-Петербург, 2007. – 928 с.

2. Миснер С. Бизнес-аналитика: Разработка куба. – «TechNet Magazine», № 11, 2009 – Режим доступа: <https://technet.microsoft.com>, свободный (дата обращения 03.05.2017)

3. Руководство по разработке для .NET Framework [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com>, свободный (дата обращения 03.05.2017).

### **С.А. СОЛОВЬЕВ**

Рязанский государственный радиотехнический университет

#### **МОДУЛЬ УПРАВЛЕНИЯ СОЛНЕЧНОЙ БАТАРЕЕЙ ДЛЯ СЛЕЖЕНИЯ ЗА СОЛНЦЕМ**

*Рассматривается новое, экономичное решение проблемы повышения эффективности солнечных панелей, которое рассчитано на работу в любых погодных условиях в дневное время суток.*

Человеку необходима энергия, причем потребность в ней с каждым годом увеличивается. В то же время запасы природных источников, таких как нефть и газ - конечны, добывать их становится всё труднее, а стоимость добычи увеличивается. В связи с этим во всем мире все шире используют альтернативные источники и в первую очередь солнце.

Преимущество солнечной энергии заключается в том, что она практически безгранична и неиссякаема, ее легко преобразовать в другие виды. Этот вид энергии экологичен и его можно получить в любой точке земного шара.

Солнечные батареи бывают нескольких типов, но широко распространены: монокристаллические и поликристаллические. Солнечные батареи с монокристаллами обладают большей рабочей эффективностью. Связано это с тем, что КПД монокристалла выше КПД поликристалла на 5% в среднем. Несмотря на то, что разница эта не слишком велика в процентном соотношении, для солнечных электростанций она может иметь решающее значение, поскольку производительность батареи должна соответствовать параметрам системы.

Кроме того, монокристаллы эффективно работают при отрицательных температурах. Поэтому если планируется использовать солнечные батареи в зимний период или круглогодично, то стоит остановить выбор именно на таком варианте. Однако поликристаллические элементы лучше зарекомендовали себя в условиях облачности и пасмурной погоды. Из-за неоднородной структуры поверхности они несколько эффективнее улавливают рассеянный свет, поэтому больше подходят для межсезонного применения. Впрочем, с развитием техно-

логий производства моноэлементов разница в падении производительности стала гораздо меньше.

**Таблица 1.**

| Фактор                  | Монокристаллы  | Поликристаллы   |
|-------------------------|--|---|
| Разница в структуре     | Кристаллы направлены в одну сторону, зёрна параллельны | Кристаллы направлены в разные стороны, не параллельны |
| Стабильность работы     | Высокая  | Меньше  |
| Стоимость               | Дорогостоящие батареи                                  | Дорогостоящие но дешевле                              |
| Окупаемость             | 2 года   | 3 года  |
| КПД                     | до 23%   | до 18%  |
| Технология производства | Совершеннее, сложнее, точнее                           | Проще, отсюда и низкая стоимость                      |

В результате сравнения основных характеристик солнечных батарей (Таблица 1), можно сделать вывод, что наиболее выгодным является использование монокристаллических батарей.

Солнечные преобразователи известны давно. Они широко распространены в мире, особенно в южных широтах. В северных широтах, где расположена большая часть нашей страны, КПД солнечных батарей существенно ниже, и он очень сильно зависит от положения панели по отношению к солнцу. Поэтому для повышения эффективности функционирования солнечной батареи необходима система ее ориентации относительно положения солнца.

Существует два основных типа поворотных механизмов для ориентации солнечных модулей: одноосевой и двухосевой.

Одноосевой реализует поворот солнечного модуля вокруг единственной центральной оси, что довольно удобно для электростанций большого масштаба. Двухосевые механизмы позволяют более гибко отслеживать положение солнца, контролируя как азимутальный, так и угол склонения солнца над горизонтом.

В статье, мной, предлагается модуль ориентации солнечной батареи, датчик угла поворота, который реализован на основе четырех фотодатчиков, направляющей и двух двигателей прямого действия с редуктором.

Конструкция проста, эффективна и имеет четыре режима работы.

Первый режим включается в том случае, когда на все четыре фотодатчика попадает солнечный свет. Это идеальное рабочее положение, при котором сбор энергии наиболее эффективен и благоприятен.

Второй режим соответствует случаю, когда один, два или три фотодатчика находятся в тени от направляющей. В этом случае начинают работать двигатели, которые отрабатывают положение солнца как по высоте, так и по азимуту до тех пор, пока все датчики света не выйдут из тени.

Третий режим - режим ожидания. Он включается в том случае, если на все датчики света не попадают солнечные лучи. Например, если солнце скрылось за тучами, модуль будет в режиме ожидания до тех пор, пока хотя бы один из датчиков не «поймает» солнечное излучение.

И четвёртый режим – возврат панели в исходную точку, в сторону восхода солнца, откуда панель начинает работу каждый новый день.

Таким образом, предложенное решение повышает точность постановки панели в перпендикулярное положение по отношению к солнцу и увеличивает общий КПД батареи.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Анализ конструкций <http://www.techsolid.ru/solievs-1004-1.html>
2. Фотодатчики и их применение <http://elektrik.info/main/praktika/600-fotodatchiki-i-ih-primenenie.html>

**М.В. ТЕЛЯЛЬКОВА, В.С. ГОРИН, В.Г. СОКОЛОВ**

Рязанский государственный радиотехнический университет

### **ОБЗОР МЕТОДОВ ПРЕОБРАЗОВАНИЯ АНАЛОГОВОГО СИГНАЛА В КОД**

*Рассматриваются методы аналого-цифрового преобразования и структуры аналого-цифровых преобразователей, используемых в системах цифровой обработки сигналов*

Для получения и обработки информации необходимо уметь принимать и обрабатывать различного рода сигналы – аналоговые, дискретные и цифровые. Параметры аналогового сигнала описываются функцией времени и непрерывным множеством возможных значений. Дискретные сигналы представляют собой отсчеты аналогового сигнала

в периодические моменты времени. В цифровых сигналах, помимо квантования по времени, присутствует квантование по уровню с представлением в виде какого-либо кода. Реализация сложных и гибких алгоритмов обработки сигнала, как правило, требует цифровой формы представления сигнала.

Устройства, реализующие функцию преобразования сигнала из аналоговой формы в цифровую, называются аналого-цифровыми преобразователями (АЦП). Суть преобразования заключается в квантовании сигнала по уровню и преобразовании непрерывного множества значений этого сигнала в дискретное множество значений уровней квантования.

Рассмотрим несколько наиболее известных методов аналого-цифрового преобразования и соответствующих видов АЦП [1, 2], преимущественно используемых в высокочастотных и широкополосных системах цифровой обработки сигналов:

- параллельного преобразования;
- последовательного приближения;
- секционированные и конвейерные.

Метод параллельного преобразования состоит в том, что входное напряжение  $U_{ax}$  с помощью блока компараторов сравнивается с рядом напряжений, формируемых из опорного напряжения  $U_{on}$  с помощью резистивной матрицы  $R$  (см. рис. 1). На выходе блока компараторов в результате возникает так называемый «термометрический код», который с помощью схемы комбинационной логики (приоритетного шифратора) преобразуется в двоичный код.

Исторически АЦП этого типа (еще не в интегральном, а в блочно-модульном исполнении) были одними из первых, нашедших применение в телевизионной и радиолокационной технике 70-х и 80-х годов. Их принципиальным недостатком является то, что количество компараторов соответствует количеству уровней дискретизации и равно  $2^m - 1$ . Ввиду сложности реализации такого количества компараторов в любом исполнении стоимость данного типа АЦП весьма высока, а разрядность, как правило, не превышает 8 бит. Однако на сегодняшний день только АЦП этого вида достигли частот дискретизации порядка 1 ГГц.

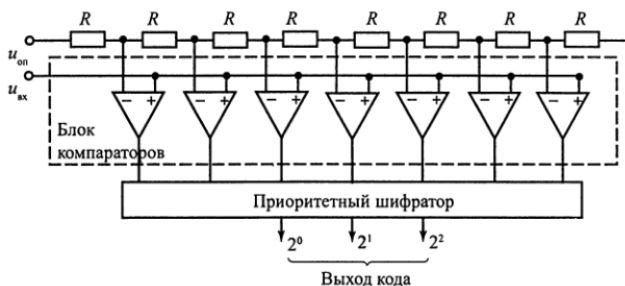


Рис. 1 – Структура АЦП параллельного преобразования.

В основе работы АЦП последовательного приближения (см. рис. 2) лежит принцип последовательного сравнения  $U_{вх}$  с значениями  $U_{он}/2$ ,  $U_{он}/4$ ,  $U_{он}/8$  и т.д. до  $U_{он}/2^m$ . Для этого работа АЦП разбивается на  $m$  тактов, в каждом из которых осуществляется сравнение  $U_{вх}$  с выходным напряжением встроенного цифро-аналогового преобразователя (ЦАП)  $U_{ЦАП}$ , причем, при  $U_{вх} > U_{ЦАП}$  в соответствующий разряд выходного кода записывается логическая «1», в противном случае – логический «0».

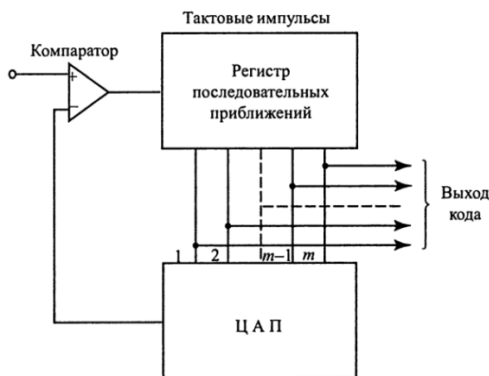


Рис. 2 – Структура АЦП последовательного приближения.

Структура данного вида АЦП проще, чем у предыдущего, однако это достигается за счет увеличения времени преобразования в  $m$  раз. Данная структура является разумным компромиссом между быстродействием, точностью и стоимостью АЦП, поэтому АЦП последовательного приближения имеют очень широкое применение в настоящее время.

Достоинства АЦП двух вышеприведенных видов частично объе-



динены в секционированном (или последовательно-параллельном) АЦП (см. рис. 3).

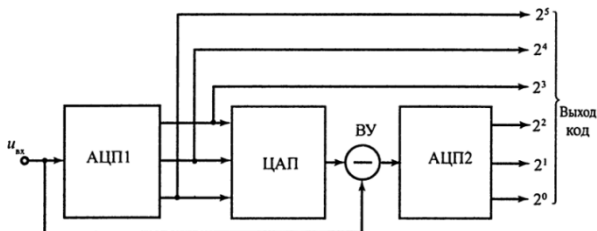


Рис. 3 – Структура 6-разрядного секционированного АЦП.

Его работа также разделена на такты, причем в первом такте осуществляется «грубое» преобразование с использованием АЦП1, формируются значения старших разрядов выходного кода и полученное значение с помощью ЦАП вычитается из  $U_{вх}$ . Результат вычитания преобразуется в младшие разряды кода с помощью АЦП2. На входе секционированного АЦП обычно присутствует схема выборки-хранения (СВХ). Достоинством такой структуры АЦП является существенно меньшее требуемое количество компараторов. Однако приходится принимать меры для выравнивания характеристик всех секций, чтобы избежать пропуска отдельных значений выходного кода.

Развитием архитектуры секционированного АЦП являются современные конвейерные АЦП, которые отличаются наличием СВХ перед каждой секцией (см. рис. 4).

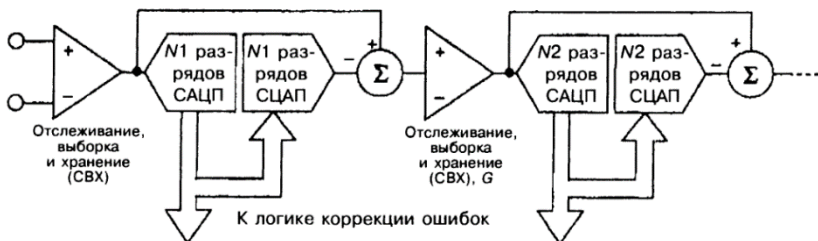


Рис. 4 – Структура конвейерного АЦП.

За счет наличия нескольких СВХ каждая секция может передавать разницу между своим входным напряжением и его преобразованным значением по «конвейеру», а на вход загружать новое значение входного напряжения для преобразования. Выходные коды старших разрядов задерживаются во вспомогательных регистрах, а ошибки, которые могут привести к пропуску кодов, корректируются соответствующей логикой. Достоинством такой архитектуры АЦП является то,

что периодичность появления достоверных значений выходного кода равна одному такту.

В заключение сравним некоторые основные характеристики АЦП приведенных видов (см. таблицу 1). По этой таблице можно выбрать наиболее приемлемый вид АЦП в зависимости от типа решаемой задачи.

**Таблица 1 – Сравнительные характеристики АЦП.**

| <b>Вид, структура</b>                     | <b>Количество разрядов</b> | <b>Время преобразования</b> | <b>Стоимость</b> |
|---|----------------------------|-----------------------------|------------------|
| Параллельного преобразования              | до 8                       | малое                       | очень высокая    |
| Последовательно-параллельные, конвейерные | до 16                      | среднее                     | высокая          |
| Последовательного приближения             | до 16                      | большое                     | средняя          |

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Аналого-цифровое преобразование/под ред. У.Кестера – М.: Техносфера, 2007 г.
2. Никамин В.А. Аналого-цифровые и цифро-аналоговые преобразователи. Справочник – М.: Альтекс-А, 2003 г.

**Е.В. ТИШКИНА**

Рязанский государственный радиотехнический университет

### **АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДЛЯ ПРОВЕДЕНИЯ МЕДИКО-ЭКОНОМИЧЕСКОЙ ЭКСПЕРТИЗЫ**

*Рассматривается автоматизация информационного взаимодействия в системе обязательного медицинского страхования отдельного региона при проведении экспертизы счетов за оказанную медицинскую помощь.*

Под автоматизацией подразумевают использование технических и программных средств, которые в некоторой мере или полностью освобождают человека от причастности к процессам приемки, обработки, передачи и использования материалов или информации. Перед процессом автоматизации происходит его формализация, то есть формирование комплексного набора указаний, имеющих однозначную

трактовку, следуя которым достигается результат реализации процесса.

Задачей информационной системы является автоматизация процесса медико-экономической экспертизы (экспертизы качества медицинской помощи), проводимого для установления соответствия фактических сроков оказания медицинской помощи, объема предъявленных к оплате медицинских услуг записям в учетно-отчетной документации медицинской организации.

Функциями разрабатываемого программного обеспечения являются:

- Импорт данных об объемах, сроках, качестве и условиях предоставления медицинской помощи в форматах, установленных приказом Федерального фонда ОМС от 07.04.2011 №79;
- Проведение медико-экономической экспертизы (экспертизы качества медицинской помощи) и отображение ее результатов удобным для дальнейшей обработки виде;
- Формирование отчетности о проведенной работе.

Разрабатываемый программный комплекс состоит из 2 модулей:

- Модуль МО – предназначен для импорта и просмотра реестров счетов в программе.
- Модуль ТФОМС – предназначен для проведения медико-экономической экспертизы (экспертизы качества медицинской помощи), для установления соответствия фактических сроков оказания медицинской помощи, объема предъявленных к оплате медицинских услуг, оказанных лицам застрахованным на территории Рязанской области, записям в учетно-отчетной документации медицинской организации.

Преимуществами автоматизации являются ускорение выполнения операций, уменьшение отказов при их выполнении, снижение издержек на реализацию и увеличение качества работы информационной системы.

Автоматизация в процессе введения в эксплуатацию и использования которой были возвращены проинвестированные в нее средства считается успешной.

Различают несколько шагов процесса автоматизации, имеющих применение в целом к роду деятельности какого-либо предприятия, каждый из которых нуждается в обдуманном и последовательном выполнении:

- 1) постановка проблемы, оценка востребованности автоматизации и возможностей предприятия;

2) формирование требований к программно-аппаратному комплексу, выбор или реализация программного продукта и технического обеспечения;

3) внедрение программного продукта;

4) обслуживание программно-аппаратного комплекса.

Перед началом автоматизации требуется точно сформулировать требования, выдвигаемые к ней. Необходимо определить функции, требующие автоматизации. Важно учесть, что как правило внедрение АИС понижает уровень воздействия человеческого фактора на процесс выполнения тех или иных операций [1].

Не стоит забывать о существующей информационной системе в автоматизируемой области если таковая имеется. Объединение в единое информационное пространство даёт возможность применять реализованные ранее компоненты и единую БД, что увеличивает гибкость ИС в общем и минимизирует дублирование информации, которая уже имеется.

В данной работе хранение данных должно осуществляться в системе управления базами данных, обеспечивающей:

- накопление, хранение и обработка данных;
- регламентация доступа к данным и управлению базой данных;
- изменение структуры и содержания данных;
- целостность данных.

В ходе анализа предметной области формируется единая модель организации, которая дает описание не только взаимодействию структурных единиц, но и выполняемым ими операциям и информационным потокам. С помощью таких документированных моделей не только оптимизируется текущая работа, но и деятельность предприятия приводится к меньшей зависимости от конкретных людей, оказать помощь в процессе подготовки новых специалистов.

Использование современных технологий позволяет не только хранить полученные данные на носителях занимающих мало места и обеспечивая при этом практически мгновенный доступ к требуемой информации, но и комбинировать и обрабатывать их различным образом для получения наглядных отчетов и графиков, расчет которых вручную был немислим из-за больших трудовых и временных затрат на него.

При создании функционального описания были рассмотрены различные аспекты, такие как программные средства, аппаратные средства, методы ввода и вывода, возможные пользователи.

Внедрение данного проекта даст следующие преимущества:

- уменьшение количества ошибок;

- значительное повышение производительности;
- удобство работы с данными;
- рационализация документооборота посредством его информатизации;
- упрощение создания, обработки и хранения документов и информации;
- повышение качества и скорости составления отчетов при снижении трудоемкости данного процесса.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем: учебное пособие.-М.: ИД «Форум»: ИНФРА-М, 2007. – 384 с.D. Savarese, «ONEWeb to Rule Them ALL». Java Pro August, 2001.

2. Карачевцева М. А., Михайлов С. М., Гуринов П. В., Савин К. А. Экспер-тиза в сфере обязательного медицинского страхования (опыт Санкт-Петербурга): журнал Вестник Росздравнадзора, Выпуск № 3 / 2013.

**А.А. ТРИШИН, А.П. БУРЯКОВ**

Рязанский государственный радиотехнический университет

### **СОВЕРШЕНСТВОВАНИЕ СИСТЕМЫ ОБМЕНА ДАННЫМИ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ**

*Рассматриваются основные способы, перспективные средства и методы повышения эффективности системы обмена данными специального назначения.*

Развитие телекоммуникационных технологий привело в последние годы к серьезным изменениям в современных инфокоммуникационных сетях.

Объединение технических средств и комплексов формирования, передачи и преобразования данных, образующих определенную структуру, которая функционирует по указанным алгоритмам, обеспечивающим передачу данными в интересах автоматизированной системы управления специального назначения (АСУ СН), представляется система обмена данными (СОД) СН.

Данные – это информация, являющиеся предметом передачи, хранения и обработки в расчетно-машинных системах. Сеть ПД подразделяется на сеть доступа (СД) и транспортную сеть связи (ТСС).

СД обеспечивает подключение ОО автоматизированных рабочих мест (АРМ), обмен сообщениями в пределах одного пункта управления и выход на ТСС. ТСС обеспечивает обмен сообщениями между различными ПУ и подключение СД. К ТСС может подключаться непосредственно и СД вынесенных автоматизированных рабочих мест (АРМ) должностных лиц (ДЛ).

Основу СД составляют абонентские комплекты устройств обмена данными (АОД).

Основу ТСС составляют узловые комплекты устройств обмена данными. Комплекты соединены каналами первичной или вторичной сети связи.

Абонентские комплекты АОД обеспечивают: прием или передачу сообщений в оконечном режиме. Имеют один абонентский стык в сторону ООД и один канальный выход. Канальный выход могут работает в двух-, четырех- или шестипроводном подключении, образуя единое информационное пространство в полудуплексном или дуплексном режимах.

Узловые комплекты АПД обеспечивают: прием или передачу сообщений в оконечном режиме или прием с дальнейшей передачей сообщения по другим каналам. Количество канальных выходов: два или более.

Все каналы нумеруются цифрами (или буквами) и всем каналам (в том числе и абонентских комплектов) присваиваются буквенный или цифровой позывной.

Комплекс технических средств ПД Т-235 предназначен для обмена и разделения данных с выдачей гарантированного засекречивания и имитозащиты.

Комплекс технических средств ПД Т-235-2.

Аппаратура обмена данными, криптозащиты и имитозащиты типа Т-236-В.

Назначение системы обмена данными.

СОД должна обладать следующими свойствами. Боевая готовность. Устойчивость функционирования, Мобильность.

Пропускная способность.  $R_{кан. ПД} = N_{ПАК} / \Delta T_{ПЕР}$  – скорость передачи данных в канале ПД может быть оценена средним количеством пакетов (или кадров), передаваемых в единицу времени (бит/с).

Разведывательная защищенность. Управляемость.

Параметры достоверности.  $P_{пр}$  – вероятность правильного приема;  $P_{ош}$  – вероятность ошибки;  $P_{ни}$  – вероятность неприема;  $P_{лт}$  – вероятность ложной тревоги.

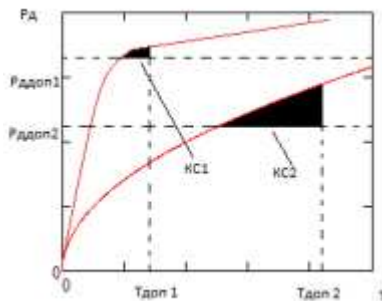
Показания вероятностей  $P_{ош}$ ,  $P_{шт}$  могут лежать в пределах от  $10^{-4}$  до  $10^{-10}$ .

Параметры своевременности.  $T_d = T_{пер} + T_{обр}$  – время доставки.

$F(t) = P_d \{T_d \leq T_{доп}\}$  – функция распределения времени доставки. Требования по своевременности доставки задаются в виде ограничений на допустимое время  $T_{доп}$  и допустимую вероятность  $P_{доп}$  доставки:  $P_d \{T_d \leq T_{доп} \mid i\} \geq P_{доп} \mid i$ . В программе *Mathcad 14* зададим следующие параметры:  $T_{доп1}=3c$ ,  $T_{доп2}=5c$ ,  $P_{доп1}=0.8$ ,  $P_{доп2}=0.6$ .

Особенности анализа эффективности функционирования сетей передачи данных. Параметры своевременности СОД СН.  $T_d = T_{пер.} + T_{обр.}$  – время доставки.  $F(t) = P_d \{T_d \leq T_{доп}\}$  – функция распределения времени доставки. Условия по своевременности доставки задаются в виде ограничений на допустимое время  $T_{доп}$  и допустимую вероятность  $P_{доп}$  доставки:  $P_d \{T_{дi} \leq T_{доп.i}\} \geq P_{доп.i}$ .

В программе *Mathcad 14* зададим следующие параметры:  $T_{доп.1} = 3c$ ,  $T_{доп.2} = 5c$ ,  $P_{доп.1} = 0.8c$ ,  $P_{доп.2} = 0.6c$ . Особенности анализа эффективности функционирования СОД СН  $F(t_d) = P(t_d) \leq t_3 \geq P_{доп}$ . Задается допустимое значение вероятности  $P_{доп}$ , при которой время, реальное, доставки  $t_d$  не превысит заданное время  $t_3$ .



**Рис. 1 – Функция распределения времени доставки с учетом заданных требований по вероятности и времени доставки сообщений для двух категорий срочности.**

Анализ этих графических зависимостей по величине заштрихованной площади показывает, что выполнить вероятностно-временные требования для доведения сообщений первой, наиболее важной, категории срочности более трудно по сравнению с доведением сообщений более низкой категории срочности.

Основные достоинства современных систем обмена данными.

1. Обмен данными по любым типам каналов связи со скоростями до 32 кбит/с.

2. Засекречивание передаваемых данных с гарантированной стойкостью и их имитозащита.

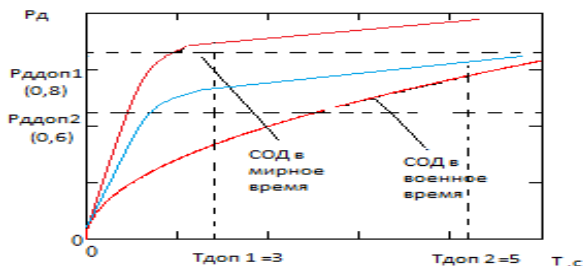
3. Выдачу информации должностным лицам на видеосистемы и средства документирования.

Недостатки современных систем обмена данными.

1. Все системы обмена данными имеют зависимость от структуры, в которой данные представлены в одной из инфокоммуникационных систем. 2. Периодичность.

3. Зависимость времени доставки сообщений от категории срочности, для передачи сообщения высшей категории, время доставки  $T_D$  значительно увеличится.

Выводы по основным свойствам СОД (параметр своевременности).



**Рис. 2 – Функция распределения времени доставки в мирное и военное время.**

В мирное время СОД имеет высокую пропускную способность ( $R_{кан.ПД}$ ), вероятность правильного приема ( $P_{пр}$ ) при сравнительно небольшом времени передачи сообщений ( $T_{доп.1} = 3c$ ) – верхняя кривая на рис. 2. Это связано с применением гражданской сети. Например, телекоммуникационная компания ОАО Ростелеком предоставляет такую возможность.



В военное время (нижняя кривая на рис. 2) параметры своевременности заметно ухудшаются, так как на КП мсбр и на КП армии развертываются штатные сети обмена данными, на основе устройств управления Т-235-1У.

При введении критерия оптимальности, который связан с числом устройств управления на командном пункте (КП), показатели своевременности повышаются, что соответствует синей кривой на рис. 2.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Титов В. С., Аверьянов Е. Г., Ащеулов С. В., Бабич Б. И., Дормидонтов А. А., Зубакин В. В., Калюка В. И., Мишин А. И., Молокович И. А., Макаров М. И., Сазонов В. В. Теоретические основы передачи данных: Уч. пособие. / Под ред. В. С. Титова – СПб.: ВАС, 2013. – 212 с.
2. Бертсекас Д., Галлагер Р. Сети передачи данных. М.: Мир, 1989. 544 с.
3. Блэк Ю. Сети ЭВМ: Протоколы, стандарты, интерфейсы. М.: Мир, 1990. 506 с

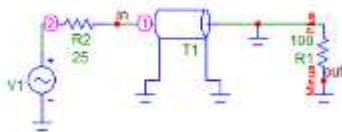
### **А.А. ТРИШИН, А.С. ЖАРОВА**

Рязанский государственный радиотехнический университет

### **ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ КАСКАДОВ С ТРАНСФОРМАТОРАМИ НА ДЛИННЫХ ЛИНИЯХ**

*Рассматриваются ряд проблемных вопросов по проектированию трансформаторов типа длинной линии (ТДЛ) в программном пакете Micro-Cap11.*

В КВ (коротковолновый), ОВЧ (диапазон очень высоких частот) диапазонах частот, применяемые для радиоустройств, свое распространение получили трансформаторы типа длинной линии (ТДЛ). У ТДЛ (Рис. 1), в отличие от обмоточных трансформаторов, верхняя частота порядка 200-250 МГц. Основным отличием от обмоточных трансформаторов в ТДЛ распространяются магнитные поля небольшой величины, что в свою очередь приводит к снижению потерь в отрезке линии и уменьшению габаритов ТДЛ. При моделировании в качестве компонента выбрали элемент «Линия передач», выполняющий функции трансформатора.



**Рис. 1 – Трансформатор типа длинной линии.**

Линия, представляющая собой отрезок определенной длины, позволяет изучать многие задачи из области применения генераторов, в частности передавать инверсные (безинверсные) сигналы. Линии необходимо согласовать, в противном случае работа ГВВ сильно ухудшается.

Для уменьшения длины отрезка(линии) как геометрической, так и электрической, отрезок(линию) наматывают, например на ферритовое кольцо (магнитопровод).

Отрезки небольшой длины проявляют некоторые достоинства, связанные с влиянием потерь(собственных). Рассогласование при этом уменьшается.

Оригинальность ТДЛ проявляется в том, что напряженность магнитного поля в ферритовом кольце вычисляется как разность токов, бегущих по проводникам. В режиме согласования - токи равны (фаза между ними 180 градусов). В результате разность токов равна нулю. Магнитопровод функционирует в режиме слабых полей, потери при этом малы.

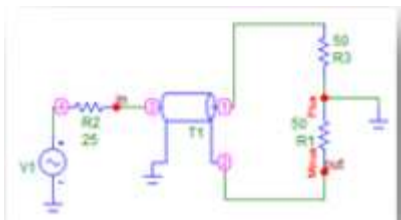
ТДЛ используют также и для разрешения проблем трансформации, путем соединения нескольких трансформаторов параллельным (при использовании трансформаторов с равными волновыми сопротивлениями), каскадным (волновые сопротивления отличаются) или параллельно-последовательным способом.

Существует несколько сочетаний “параллельного” способа:

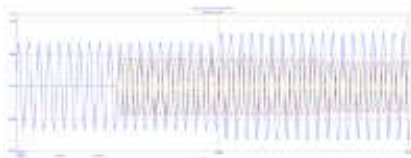
- 1) входы ТДЛ соединены параллельно, а выходы последовательно(повышающий ТДЛ);
- 2) входы последовательно, выходы параллельно(понижающий ТДЛ).

Исследуя собранные схемы, а именно осуществление перехода от несимметричного генератора к симметричной нагрузке (Рис. 2), было выявлено, что особенностью такого моделирования является задержка установления процессов во времени (Рис. 3), то есть моделирование происходит с какой-то периодичностью .

Это объясняется тем, что коэффициент передачи проводника - комплексная величина. Анализ переходных процессов производится при помощи интеграла свертки импульсной характеристики, вычисление которой происходит с использованием преобразования Фурье.



**Рис. 2 – Переход от несимметричного генератора к симметричной нагрузке.**



**Рис. 3 – Задержка установления процессов во времени.**

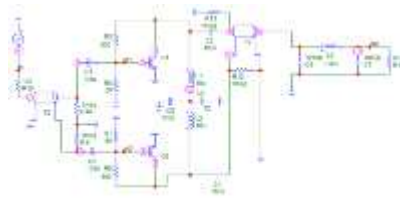
При моделирование ТДЛ был обнаружен существенный недостаток. Невозможно проводить анализ амплитудно-частотной характеристики (Рис. 4)



**Рис. 4 - Анализ АЧХ трансформатора типа длинной линии.**

Также в данной работе проводились исследования широкодиапазонного генератора на основе трансформатора типа длинная линия (Рис. 5).

Широкодиапазонные генераторы гарантируют гармоническое напряжение или ток в апериодической нагрузке. Вместе с этим входное сопротивление обладает высокой линейностью. Опасность перехода транзистора в перенапряженный режим минимальна.



**Рис. 5 – Широкодиапазонный генератор.**

При рассчитанных значениях элементов, входящих в электрическую схему, мощность на выходе данного генератора составляет 92 мВт

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Шахгильдян В.В., Козырев В.Б., Ляховкин А.А., Нуянзин В.П., Розов В.М., Шумилин М.С. Радиопередающие устройства: Учебник для вузов; Под ред. В.В. Шахгильдяна. – 3-е изд., перераб. и доп.-М.: Радио и связь, 2003.- 560с.: ил.
2. Белов Л.А., Богачев В.М., Благовещенский М.В. и др. Устройство генерирования и формирования радиосигналов; Под ред. Г.М. Уткина, В.Н. Кулешова и М.В. Благовещенского.- М.: Радио и связь, 1994.-416с.
3. Шумилин М.С., Козырев В.Б., Власов В.А. Проектирование транзисторных каскадов передатчиков. Учеб.пособие для техникумов.- М.:Радио и связь, 1987. - 320с.: ил.
4. Гоноровский И.С., Демин М.П.. Радиотехнические цепи и сигналы.- М.: Радио и связь, 1994.- 480с.

**А.М. ФАМ, А.П. ШИБАНОВ**

Рязанский государственный радиотехнический университет

### **РАЗРАБОТКА ПРОГРАММЫ ИМИТАЦИОННОЙ МОДЕЛИ СМО М/М/1**

*Рассматриваются теоритические и практические вопросы моделирования имитационной модели СМО М/М/1.*

**Введение.** С системами массового обслуживания (СМО) мы встречаемся повседневно. В банке или в магазине, кафе и т.д., мы можем видеть клиентов, стоящих в очереди и ожидающих обслуживания. Это простой пример системы массового обслуживания. Более того,

любое производство можно представить как последовательность систем обслуживания. Из изучения СМО и из полученных результатов приводятся оценки работоспособности системы, для чего нам необходимо моделировать систему.

**Целью работы** является разработка приложения, цель которого заключается в создании программы, моделирующей имитационной модели СМО М/М/1. Результаты моделирования используются для проведения оценки эффективности модели и на их основе могут быть получены практические рекомендации.

**Теоретическая часть.** Система состоит из множества взаимосвязанных элементов, объединенных для выполнения определенной функции. Модель представляет собой абстрактное описание системы. Разработка модели - сложный процесс, который во многом является искусством, но, однако, он упрощается, если: 1) известны физические законы, описывающие функционирование системы; 2) может быть разработано графическое представление системы; 3) можно управлять входами, элементами и выходами системы[2].

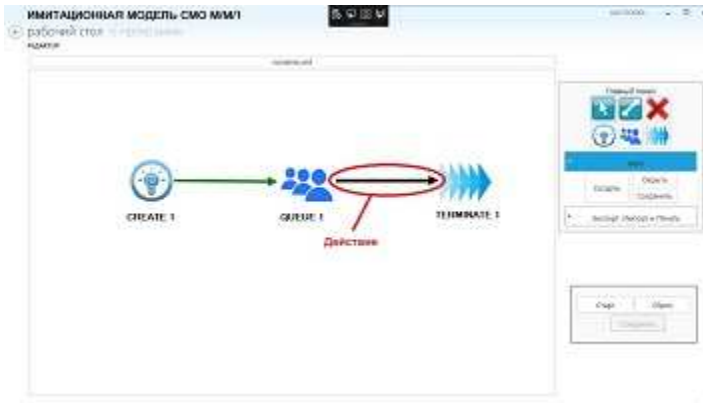
Для построения имитационной модели, сначала надо определить цели разработки модели, на этой основе устанавливаются границы и уровень детализации модели. Модель составляется в соответствии с критериями эффективности и альтернативными решениями. После получения рекомендации можно приступить к реализации результатов. С помощью имитационного моделирования могут строиться как агрегированные, так и детализированные модели[1].

Для обозначения СМО используются три переменные для первых трех параметров:  $X/Y/Z$ , где  $X$  – распределение времени поступления;  $Y$  – распределение времени обслуживания;  $Z$  – число обслуживающих устройств. Самая известная модель СМО является типом М/М/1. Это система с одним устройством обслуживания, и где буква М означает марковские процессы распределения времени поступления и обслуживания. Время между поступлениями в систему требований и время обслуживания имеют экспоненциальные распределения.

При моделировании СМО, необходимо описать их характеристики: входящий поток требований, которые поступают на устройство обслуживания; дисциплину постановки требований в очередь и выбор из нее; правило, по которому осуществляется обслуживание; выходящий поток требований.

#### **Имитационное моделирование.**

На рис. 1 приведено компьютерное изображение имитационной модели СМО М/М/1. В программе содержатся узлы и дуги, которые предназначены для разных функций.



**Рис. 1 – Имитационная модель СМО М/М/1.**

Узел CREATE используется для генерирования заявок и отправки их в систему. Время возникновения первой заявки в узле задается переменной TF, а время между возникновением следующих за первой заявкой – TBC. Максимальное число заявок, возникающих в данном узле – MC.

Узел – очередь QUEUE, предназначен для определения места, где заявки ожидают обслуживания. Когда заявка поступает в узел, её место расположения в очереди определяется статусом обслуживания, принятым для данного узла. Если устройство обслуживания занято, заявка ожидает в узле до того момента, когда начнется её обслуживание. Если устройство не занято, заявка сразу обслуживается. После освобождения обслуживающего устройства заявка автоматически извлекается из очереди и инициируется следующее обслуживание. Для узла QUEUE определяются IQ – начальное число заявок в узле и QC – емкость очереди, т. е. максимальное число заявок, которые могут ожидать обслуживания в узле.

Для моделирования действий используются дуги. Действия, начинающиеся в узлах QUEUE, называются обслуживающими действиями. Каждая дуга имеет начальный и конечный узлы. При определении маршрута вывода заявки из начального узла необходимо выбрать дугу, по которой он будет следовать. Действие имеет ещё параметры и способы для выбора действия:

- Продолжительность действия, оно может задаваться с помощью любого выражения. Например, продолжительность может быть равна значению выборки из экспоненциального распределения с математическим ожиданием и т.д.

- Выбор может осуществляться вероятностным способом, и тогда значения вероятностей являются частью описания действия. Приписываемые дугами вероятности являются вещественными числами, лежащими в пределах от 0,0 до 1,0. Сумма вероятностей, приписанных вытекающим из одного и того же узла дугам, должна быть равна единице.
- При условном выборе частью описания действия являются введенные условия. Однако для обслуживающих действий условия не задаются. Условие является одной из стандартных операций отношения в Фортране (LT - меньше, LE - чем; меньше, чем или равно; EQ - равно; NE - не равно; GT - больше, чем; GE - больше, чем или равно).

Узел TERMINATE используется для задания числа заявок, по окончании обработки которых завершается имитационный прогон, и для уничтожения или вывода заявок. С помощью узла TERMINATE можно задать условие, по которому для завершения прогона моделирования требуется, чтобы в этот узел прибыло TC заявок.

После моделирования определяются следующие параметры: средняя длина очереди; максимальная длина очереди; число входов заявок; число нулевых входов; среднее время пребывания в очереди; среднее время пребывания в очереди с учетом нулевых входов; диаграммы времени пребывания в очереди и текущей длины очереди. Примеры диаграмм приведены на рис. 2-3:

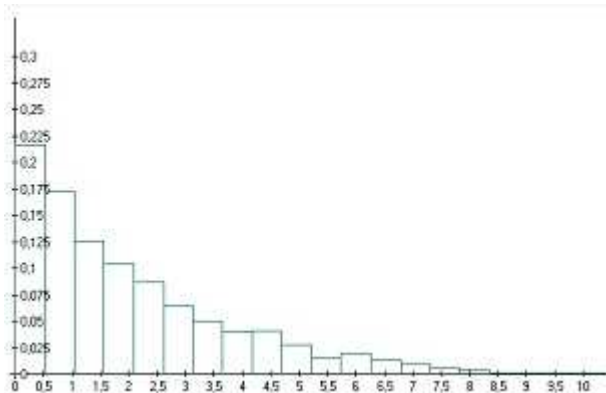
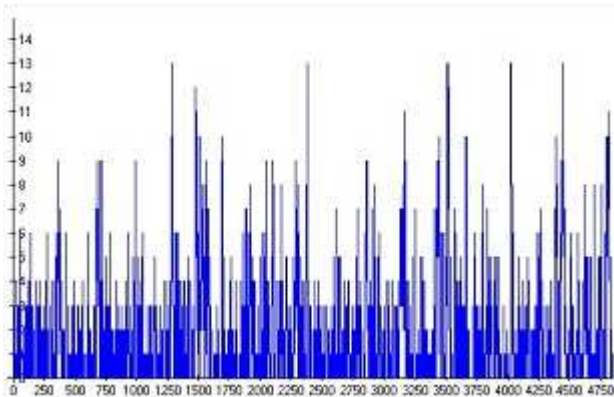


Рис. 2 – Пример диаграммы времени пребывания в очереди.



**Рис. 3 – Пример диаграммы текущей длины очереди.**

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II: Пер. с англ. - М.: Мир, 1987. - 646 с.
2. Pritsker, A. A. B., Modeling and Analysis Using Q-GERT Networks, Halsted Press and Pritsker & Associates, Inc., 1977.

**В.Т.ФАМ, В.В. СУСКИН**

Рязанский государственный радиотехнический университет

### **ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ ФОТОЭЛЕКТРИЧЕСКОЙ ТЕПЛОЙ ПАНЕЛИ**

*Предлагается способ повышения эффективности фотоэлектрической панели, функционирующей в условиях повышенной температуры. Рассматриваются особенности способа и расчет коэффициента полезного действия фотоэлектрической тепловой панели.*

В настоящее время в регионах с наиболее высокой солнечной инсоляцией 1800 кВт час в год на 1 м<sup>2</sup> и более практически отсутствуют солнечные установки прямого преобразования солнечной энергии в электрическую (фотовольтаические СЭУ, на базе солнечных фотоэлектрических панелей).

Такое положение дел вызвано двумя причинами, связанными с высокой температурой окружающей среды в этих регионах:

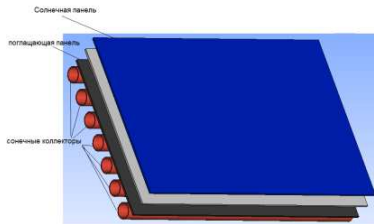


- при температуре окружающей среды около  $+45^{\circ}\text{C}$  собственная температура ФСМ становится выше температуры  $+85^{\circ}\text{C}$  – максимально допустимой гарантийной температуры работы модуля;

- при высокой температуре окружающей среды резко, на 30-40 %, падает мощность ФСМ, что делает проекты по созданию фотовольтаических СЭУ малоэффективными.

Одним из известных способов, при котором уменьшается температура солнечной панели, является фотоэлектрическая панель, использующая радиатор с проточной водой, или фотоэлектрическая тепловая панель. В данной работе рассматривается особенность функционирования фотоэлектрической тепловой панели.

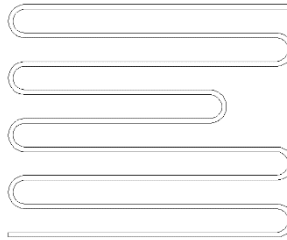
Как правило, тепловые и фотоэлектрические системы используются отдельно. Было показано, что эти системы могут быть объединены, чтобы сформировать комбинированную систему, называемую фотоэлектрической тепловой системой или фотоэлектрической тепловой панелью, использующей радиатор с проточной водой, представлена на рис. 1.



**Рис. 1 – Конструкция фотоэлектрической тепловой панели, использующей радиатор с проточной водой.**

В данной конструкции фотоэлектрической тепловой панели на обратной стороне расположен радиатор с проточной водой для охлаждения активного элемента преобразующего солнечные энергии в электричество. Движение проточной воды в радиаторе охлаждения осуществляется либо с применением модуля ориентации на солнце, либо естественной циркуляцией воды за счет разницы температур верхней и нижней части радиатора солнечной панели.

В рассмотренной фотоэлектрической панели происходит преобразование тепловой и электрической энергии в электричество. фотоэлектрические панели, в которых комбинируют тепловые и электрические энергии, вырабатывают не только электричество, но и тепло, называется фотоэлектрической тепловой системой (ФТС).



**Рис. 2 – Радиатор с проточной водой.**

Рассчитаем КПД фотоэлектрической тепловой панели.

Фотоэлектрический КПД фотоэлектрической тепловой панели определяется по формуле [4]:

$$\eta_e = \frac{I_{\max} \cdot U_{\max}}{G \cdot A} \quad (1)$$

Где,  $\eta_e$  - Фотоэлектрический КПД солнечной панели.

$I_{\max}$  - максимальный ток солнечной панели.

$U_{\max}$  - максимальное напряжение солнечной панели.

$G$  – коэффициент поглощения солнечной радиации (W/m<sup>2</sup>).

$A$  – площадь солнечного модуля (m<sup>2</sup>).

Тепловое КПД солнечного модуля определяется по формуле:

$$\eta_t = \frac{m \cdot C \cdot (T_2 - T_1)}{G \cdot A} \quad (2)$$

Где,  $\eta_t$  - Тепловое КПД солнечного модуля.

$m$  – масса воды внутри радиатора.

$C$  - удельная теплоемкость воды (4200 J/Kg.K).

$T_2$  - температура на выходе радиатора.

$T_1$  - температура на входе радиатора.

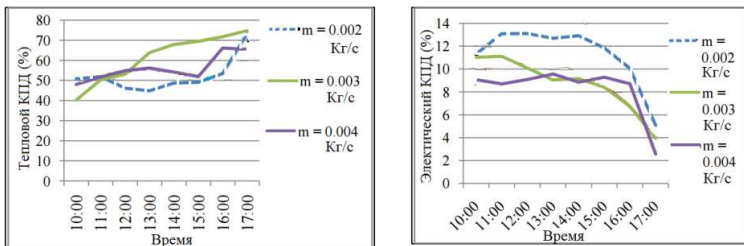
Общий КПД фотоэлектрической тепловой панели рассчитывается на формулу:

$$\eta_o = \eta_e + \eta_t \quad (3)$$

$$\eta_o = \frac{I_{\max} \cdot U_{\max}}{G \cdot A} + \frac{m \cdot C \cdot (T_2 - T_1)}{G \cdot A} \quad (4)$$

$$\eta_o = \frac{I_{\max} \cdot U_{\max} + m \cdot C \cdot (T_2 - T_1)}{G \cdot A} \quad (5)$$

Из уравнения (5) следует, что общий КПД фотоэлектрической тепловой панели зависит от характеристик жидкости, которая протекает внутри трубы, например, от удельной теплоемкости жидкости, температуры жидкости на входе трубы, массы жидкости, т.е. скорости потока жидкости. В результате экспериментального исследования показаны, что КПД фотоэлектрической тепловой панели снижается на 0,5% при повышении температуры окружающей среды на  $1^{\circ}\text{C}$  [4], график зависимости КПД фотоэлектрической тепловой панели от скорости потока жидкости, представлен на рис. 3.



**Рис. 3 – Зависимость КПД фотоэлектрической тепловой панели от скорости потока жидкости.**

### Заключение

В данной статье представлена фотоэлектрическая тепловая панель, использующая радиатор с проточной водой. В условии повышенной температуры температура панели снижается с помощью радиатора с проточной водой, при этом приводит к повышению КПД фотоэлектрической панели. Кроме этого, система вырабатывает горячую воду. В результате экспериментального исследования определена зависимость КПД фотоэлектрической тепловой панели от различных значений скорости потока жидкости.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Джумаев А. Я., анализ влияния температуры на рабочий режим фотоэлектрической солнечной станции / Технические науки - от теории к практике: сб. ст. по матер. XLVI междунар. науч.-практ. конф. № 5(42). – Новосибирск: СибАК, 2015.
2. Джумаев А. Я., анализ влияния температуры на рабочий режим фотоэлектрической солнечной станции / Технические науки - от теории к практике: сб. ст. по матер. XLVI междунар. науч.-практ. конф. № 5(42). – Новосибирск: СибАК, 2015.

3. Безручко К. В., Расчет характеристик солнечных батарей на основе математических моделей / Учебное пособие для курс. И дипл. проект. – Харьк. Авиац. Ин-т., Харьков, 1995.

4. Huang BJ et.al. Performance evaluation of solar photovoltaic/thermal systems. Sol Energy 2001;70;443-8

### **В.В. ФИЛИМОНОВ, В.В. СУСКИН**

Рязанский государственный радиотехнический университет

### **ЭНЕРГОСБЕРЕГАЮЩИЕ ОСВЕТИТЕЛЬНЫЕ ПРИБОРЫ ДЛЯ АВТОМОБИЛЕЙ С ЭЛЕМЕНТАМИ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ**

*Рассматриваются современные решения проблемы безопасности дорожного движения в части освещения и комфорта водителей при управлении автомобилем в темное время суток.*

Одна из наиболее значимых проблем безопасности дорожного движения – освещение. По статистике, за 2016 год 12% ДТП в России произошло по причине, в той или иной мере, связанной с рассматриваемой нами проблемой. Конечно, следует понимать, что проблема комплексная и решать ее требуется на глобальном уровне, помимо освещения на безопасность влияют: дорожное полотно, разметка, физическое и моральное состояние водителя, исправность транспортного средства и многое другое. Исследуя проблему освещения, можно отметить недостаток штатного освещения дорог, а зачастую даже его отсутствия за пределами города. В этом случае автомобилисты вынуждены полагаться только на головное освещение своего автомобиля, через которое я и предлагаю решить данную проблему.

Проанализировав существующие решения, было выявлено, что в бюджетном классе автомобилей, наиболее часто применяется замена галогенных ламп на ксеноновые или светодиодные.

Таблица 1.

|                       | Галогенные | Ксеноновые  | Светодиодные |
|-----------------------|------------|-------------|--------------|
| Потребляемая мощность | 45-60 Вт   | 35 Вт       | 25 Вт        |
| Цветовая температура  | 3300 К     | 4300-8500 К | до 7000 К    |
| Световой поток        | 1700 лм    | 3000 лм     | 3600 лм      |
| Срок службы           | 1000 ч.    | 2000 ч.     | 50000 ч.     |
| Стоимость за комплект | 500 р.     | 1000 р.     | 4000 р.      |

В результате сравнения основных характеристик ламп (Таблица 1), можно сделать вывод, что при одинаковом показателе освещенности в соответствии со стандартом, действующим на территории нашего государства (*РФ ГОСТ Р 41.1-99*), наиболее выгодным является использование светодиодных ламп. Несмотря на то, что их стоимость в разы дороже чем у аналогов, в долгосрочном периоде затраты на покупку новых ламп становятся меньше. Благодаря небольшой потребляемой мощности мы получаем экономию на электроэнергии, а как следствие экономиию на топливе.

Не менее важным показателем при выборе ламп для головного освещения автомобиля является цветовая температура, так как от нее будет зависеть точность и комфорт восприятия водителем окружающего пространства. Как вы можете знать, усредненная цветовая температура дневного света равна 6000К и хорошо воспринимается нашими глазами в повседневной жизни. Несмотря на это, результаты экспериментов с автомобильным освещением дают нам немного другую информацию, причина этого в том, что во время движения мы сталкиваемся с разными погодными условиями, таким как дождь и туман. С учетом этого получается, что в сухую погоду наиболее хорошо себя показывают лампы, свет которых приближен к дневному. В пасмурную погоду, а именно в момент выпадения осадков, излучаемый фарами дневной свет практически невиден, что объясняется большим количеством отражений. Универсальным вариантом стали лампы, цветовая температура которых находится в диапазоне от 3800 до 4300К, что соответствует желтоватому свету. Данные параметры наилучшим образом можно реализовать при помощи светодиодов.

В дорогом сегменте автомобилей используются более сложные технологические решения. Например, фирма Audi начиная с седанов шестой модели, применяет так называемую матричную оптику с авто-

матическим управлением света. Головная оптика автомобиля состоит из ряда светодиодных линз, а автоматическое управление светом позволяет водителю полностью забыть о переключении дальнего света на ближний, так как при приближении встречного автомобиля система сама регулирует яркость светодиодов, чтобы не ослеплять водителей. Система такого типа обеспечивает максимально допустимое освещение на протяжении всего пути, что позволяет водителю лучше сосредоточиться на управлении транспортным средством и избежать дополнительной утомленности.

По состоянию рынка, можно сказать, что тема осветительных приборов для автомобилей пользуется большим спросом и наблюдается тенденция постепенного роста. Выявив отсутствие высокотехнологичных решений в бюджетном сегменте авто мы начали проектирование собственного. Оно будет комплексное, и состоять из нескольких частей, которые, как мы считаем, смогут существовать на рынке в виде самостоятельных продуктов.

Одним из наших решений будет создание собственного конструктивного решения на базе лампы H4 с цоколем E39, его особенностью будет наличие стабилизатора тока, что делает лампу универсальной, и дает возможность для подстройки уровня света под конкретный автомобиль в соответствии с ГОСТ Р 41.1-99.

Вторым техническим решением является автоматическое управление светом. Задача системы будет сводиться к простому переключению дальнего света на ближний при приближении встречного автомобиля, а затем включении обратно, тем самым водитель сможет быть более сосредоточен на управлении транспортным средством. Уникальность этой системы заключается в том, что она рассчитана на бюджетные авто, и поэтому при ее проектировании основной задачей было сделать ее дешевой. Следовательно, в отличие от той же системы Audi, в нашей разработке не предполагается использование сложного программного обеспечения, которое могло бы корректировать свет по несколько люмен и перемещение самого диода. Установка такой системы на автомобиль подразумевает подключение дополнительного устройства в цепь между блоком управления фарами и элементами освещения. В качестве вариантов дальнейшего развития, мы рассматриваем возможность разработки закрытого модуля фары под типоразмеры конкретного автомобиля, который будет включать в себя весь комплекс наших устройств.

Вернувшись к статистике, стоит отметить, что 42% всего населения нашей страны являются автовладельцами и это количество увеличивается в среднем на 2% за год. По сведениям из различных источ-

ников 50-65% владеют Российскими автомобилями, что доказывает актуальность проблемы именно для бюджетного сегмента.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Юшин А.М. "Современные светодиоды". 2013.
2. News.drom.ru
3. Государственный стандарт РФ ГОСТ Р 41.1-99 (Правила ЕЭК ООН N 1) "Единообразные предписания, касающиеся официального утверждения автомобильных фар, дающих асимметричный луч ближнего и (или) дальнего света и оснащенных лампами накаливания категории R<sub>2</sub> и (или) HS1" (принят и введен в действие постановлением Госстандарта РФ от 26 мая 1999 г. N 184).

### А.С. ФРОЛОВ

Рязанский государственный радиотехнический университет

### СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАБОТЫ С ЕДИНОЙ ГОСУДАРСТВЕННОЙ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМОЙ

*В данной статье рассматриваются существующие прикладные решения для работы с ЕГАИС, а также проводится сравнительный анализ разрабатываемого решения с существующими.*

Единая государственная автоматизированная информационная система (ЕГАИС) предназначена для государственного контроля за производством и оборотом различной продукции (лесозаготовки, меховые изделия, алкогольная продукция и пр.). Система предоставляет единый интерфейс для всех участников рынка и обязывает их оперативно предоставлять данные о всех хозяйственных операциях, связанных с данной продукцией.

Однако программные средства, предоставляемые государством, (УТМ – универсальный транспортный модуль) не имеют графического пользовательского интерфейса (GUI) как такового и управляются программно посредством приёма-передачи XML-файлов с данными фиксированной структуры. Формат данных закреплён законодательно и подробно описан в нормативно-правовых актах.

В связи с тем, что текстовое представление данных (даже в виде XML) не всегда удобно для визуального восприятия, за последние два

года на рынке появилось множество программных решений и программно-аппаратных комплексов для работы с функционалом ЕГАИС.

Среди программных продуктов следует отметить такие, как:

1. «МойСклад» - торговая система с интегрированным CRM и поддержкой ЕГАИС [1]

2. «MasterUTM» - программное обеспечение от разработчика ЕГАИС - АО «Центринформ» [2]

3. Обновления существующего встроенного ПО для POS-терминалов. (POS – point of sale – торговое оборудование)

Среди программно-аппаратных комплексов следует выделить следующие:

1. Комплекты Viki (Дримкас) [3] - включают в себя:

а. Сенсорный моноблок с диагональю экрана 7”

б. Кассовая программа с ограниченным функционалом

с. Дополнительно приобретаются фискальный регистратор, сканер штрихкодов, денежный ящик, дисплей покупателя.

2. Автономная касса ЕГАИС (АТОЛ) [4] – включает в себя:

а. УТМ АТОЛ HUB-19 – мини-компьютер на базе ОС Linux с предустановленным фирменным ПО

б. АТОЛ 90Ф ЕГАИС – фискальный регистратор, соответствующий требованиям законодательства касательно работы с ЕГАИС и передачи данных в налоговую инспекцию (54ФЗ)

с. Honeywell 1450g – сканнер штрихкодов с поддержкой стандарта PDF417.

3. Смарт-терминал «Стандарт ФН Плюс» (Эвотор) [5] – включает в себя:

а. Смарт-терминал – планшетный компьютер на базе UNIX-подобной ОС «Эвотор»

б. Фискальный регистратор

с. 2D-сканнер

Предметом моей научно-исследовательской работы также является разработка прикладного ПО для работы с системой ЕГАИС, получившее название «РИЛ: ЕГАИС». Данное ПО представляет собой конфигурацию БД для платформы 1С:Предприятие 8.3. Пользователи осуществляют доступ к БД посредством веб-клиента.

По сравнению с перечисленными продуктами, разрабатываемое ПО предлагает аналогичные возможности, но в то же время отличается рядом преимуществ. Так, например, стоит отметить, что «РИЛ: ЕГАИС» имеет низкие требования к аппаратному и программному обеспечению. Аппаратные требования ограничиваются лишь теми, что



обозначены для установки УТМ – для работы с «РИЛ: ЕГАИС» достаточно лишь наличия веб-браузера, который в большинстве случаев является предустановленной частью ОС. Таким образом, для установки «РИЛ: ЕГАИС» подойдет любой ПК, на который возможна установка УТМ и в этом случае не требуется приобретение отдельных дорогостоящих аппаратных решений, как в случае комплектов Дримкас, АТОЛ или Эвотор.

Также, «РИЛ: ЕГАИС» не выдвигает дополнительных требований к скорости Интернет-соединения. Объем загружаемой части клиентского приложения и передаваемых данных невелики, что делает возможной работу даже при доступе к сети Интернет по технологиям 3G или ADSL. Для сравнения, «Автономная касса ЕГАИС», периодически получает обновления всего образа предустановленной ОС размером порядка 700Мб, что может приостановить работу на длительное время при низком качестве Интернет соединения.

Обновления «РИЛ: ЕГАИС» происходят централизованно и не требуют от пользователя дополнительных действий или навыков. Серверная часть приложения в некоторых случаях может быть обновлена динамически без прерывания работы пользователя. Если же обновление относится к структуре БД, то работа пользователей будет завершена с предварительными уведомлениями каждую минуту за 10 минут до начала работ. Сам процесс обновления обычно занимает не более 10 минут и его длительность зависит только от объемов информационной базы, что существенно повышает доступность сервиса, по сравнению с другими решениями.

Уникальной особенностью системы «РИЛ: ЕГАИС» является наличие централизованной базы торговых операций, позволяющей осуществлять встречные сверки между различными контрагентами и исключить возможность дублирования регистрации продаж т.н. «задвоенных» акцизных марок (контрафактной продукции), что в конечном итоге позволяет конечному пользователю отлеживать легальность реализуемой продукции.

Среди прочих особенностей необходимо также выделить простоту и интуитивность пользовательского интерфейса. Информативность интерфейсных элементов позволяет начать пользоваться программой без предварительной подготовки или обучения. Кроме того, интерфейс оснащен встроенными подсказками, а также подробным руководством пользователя, доступным непосредственно из программы.

Архитектурное решение, выбранное для «РИЛ: ЕГАИС» позволяет также выделить еще одну особенность, которая на данный момент

не реализована ни в одном из конкурирующих решений, а именно централизованное взаимодействие со всеми обособленными подразделениями в рамках группы из единого интерфейса. Пользователь может самостоятельно выбрать необходимое подразделение и начать с ним работу при условии установленного Интернет-соединения. Кроме того, применяемая архитектура позволяет оперативно включать/отключать отдельные подсистемы и функциональные опции средствами администратора системы, что делает систему гибкой и позволяет использовать модель обслуживания «Software&support as service (SaaS)» - «ПО как услуга» или «Software on demand» - «ПО по требованию».

Использование такой модели обслуживания позволяет оперативно получать отзывы пользователей о работе программы, выявлять и устранять возникающие ошибки, а также получать пожелания пользователей по расширению функционала, что укладывается в итеративную модель разработки ПО.

Использование клиент-серверной архитектуры позволяет также организовать более тесное взаимодействие с пользователем. Так, например, в «РИЛ: ЕГАИС» применяется подсистема гибких уведомлений и напоминаний, которые могут быть динамически настроены администратором системы и становятся доступны пользователю в режиме реального времени.

Единственным недостатком такого подхода является требование наличия постоянного доступа к сети Интернет при работе с приложением. Однако данное требование совпадает с требованиями законодательства, так как нормативно-правовые акты предписывают регистрировать хозяйственные операции и передавать данные о них на сервера государственных служб либо в режиме реального времени, либо в достаточно ограниченные сроки.

Ещё одной уникальной особенностью «РИЛ: ЕГАИС», которой необходимо уделить внимание, является функция выгрузки данных в формате «Enterprise Data», что позволяет осуществлять обмен информацией с широко распространенными прикладными решениями фирмы 1С. Такой обмен данными позволяет существенно облегчить работу бухгалтерских отделов пользователей, избавив их от повторного внесения данных в учетные системы.

### **Заключение**

В данной статье приведен обзор программных и программно-аппаратных решений для работы с единой государственной автоматизированной информационной службой (ЕГАИС). На момент написа-

ния статьи, рынок изобилует различными решениями, отличающимися как ценовой политикой, так и набором доступных опций.

Разрабатываемая система «РИЛ: ЕГАИС» не только максимально широко охватывает существующие требования пользователей, но и предоставляет максимально комфортную работу, не требуя от пользователя дополнительных знаний и навыков. Также, «РИЛ: ЕГАИС» обладает рядом возможностей, которые не могут предоставить конкурентные решения.

Кроме того, система легко масштабируема благодаря увеличению вычислительной мощности сервера приложения и использования технологии кластеризации, что обеспечивает комфортное быстродействие системы не зависимо от числа пользователей.

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. «Складской учет товаров онлайн | Мой склад» - <https://www.moysklad.ru/>
2. «Специализированное ПО | АО ЦентрИнформ» - [https://www.center-inform.ru/egais/oborudovanie/spetsializirovannoe-po/#po\\_masterutm](https://www.center-inform.ru/egais/oborudovanie/spetsializirovannoe-po/#po_masterutm)
3. «Автоматизация торговли в розничном магазине - Дримкас» <https://dreamkas.ru/>
4. «АТОЛ Автономная касса ЕГАИС» - [http://www.egais.atol.ru/products/complex/solutions\\_27.html](http://www.egais.atol.ru/products/complex/solutions_27.html)
5. «Терминалы - Эвотор» - <https://evotor.ru/terminals/>

### Е.Ю. ХОЛОПОВ

Рязанский государственный радиотехнический университет

### ПОДАВЛЕНИЕ СЛУЧАЙНЫХ ГЕОМЕТРИЧЕСКИХ ДЕФОРМАЦИЙ В РЕАЛЬНОМ ВРЕМЕНИ НА БАЗЕ ПЛИС

*В данной статье рассматриваются существующие группы методов, используемые для обнаружения движущихся объектов.*

Компьютерные системы обработки и анализа видеoinформации всё более интенсивно применяются в различных областях человеческой деятельности. Наиболее широкое применение они получили в таких задачах как навигация, космический мониторинг Земли, контроль качества и количества производимой продукции, обеспечение безопасности различных объектов, передача и хранение видеoinформации, медицинские и военные приложения.

Одним из направлений при создании систем анализа и обработки видеoinформации является разработка бортовых видеoinформационных комплексов, которые предназначены для установки на автономных носителях, таких как самолёты, вертолёт, автомобили. Характерными чертами этих бортовых систем технического зрения являются, во-первых, необходимость работы в реальном времени. Во-вторых, эти системы должны иметь автономный режим функционирования, при котором требуются лишь незначительные усилия оператора.

Одной из наиболее актуальных проблем, связанных с разработкой бортовых систем технического зрения, является проблема обнаружения движущихся объектов. Обнаружение движущихся объектов находит применение не только в военных системах, но и в системах поиска и спасения, охранных комплексах, системах контроля выполнения правил дорожного движения на дорогах. Кроме того, обнаружение часто лежит в основе алгоритмов распознавания объектов и понимания сцен.

Задача обнаружения объектов состоит в принятии решения о наличии или отсутствии объекта, или нескольких объектов в текущем изображении.

Всё многообразие методов, используемых для обнаружения, можно разделить на пространственные и временные, хотя использование комбинированных методов приносит гораздо более точные результаты.

Пространственная обработка базируется на информации, заложенной в одном телевизионном кадре. Она так же учитывает корреляционные связи между выборками сигналов в соседних точках изображения.

Временная обработка основана на заложенной в нескольких телевизионных кадрах информации, и учитывает корреляционные связи между выборками сигналов в одной и той же точке изображения в различных кадрах. [1]

Первая группа методов проводит независимую обработку внутри отдельных телевизионных кадров с целью обнаружения на них всех объектов с заданными признаками. После происходит межкадровая обработка с использованием анализа динамики координат объектов. Она позволяет отделить подвижные объекты от неподвижных. В случае отсутствия полной информации об объекте, обнаружение проводят на основе априорной информации о геометрических, яркостных, спектральных, текстурных, статистических и других признаках.

Вторая группа методов обнаружения объектов основана на межкадровой обработке телевизионных кадров с последующей внутрикадровой обработкой. Поскольку методы обнаружения объектов на сложном фоне, базирующиеся на временной обработке сигналов, при обнаружении дополнительно используют информацию, получаемую в предыдущих кадрах, эффективность таких методов, безусловно, выше, чем у методов, использующих при обнаружении только априорную информацию о свойствах сложного фона. [2]

В случае, если перемещения датчика изображений являются непредсказуемыми, то нельзя предполагать наличие какой-либо связи между соответствующими точками соседних кадров, поэтому применение алгоритмов с временной обработкой практически невозможно, исключая случай ровного фона. [3]

Таким образом, использование пространственно-временного подхода для обнаружения движущихся объектов требует знания пространственного соответствия между точками различных кадров. Если датчик неподвижен, то это соответствие тождественно. Если известен закон перемещения датчика изображения и трёхмерная модель наблюдаемой сцены, то мы сможем восстановить это соответствие на основании математической модели формирования изображения. [4]

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. А.С. 1427396 СССР. Устройство для определения положения объекта на изображении / Б.А. Алпатов, А.А. Селяев, С.Ю. Хлудов. Опублик. 30.09.88. Бюл. № 36, 9 с.Э. Ньюкомер, Веб-сервисы: XML, WSDL, SOAP и UDDI, Питер, 2003г.
2. Применение методов сегментации изображений в автономных системах обнаружения, распознавания и сопровождения движущихся целей / Под ред. П. А. Бакута // Зарубежная радиоэлектроника. – 1987. – №10. – С. 3 – 93.
3. Бачило С.А., Дзягун Д.Ю., Итенберг И.И., Овчаров А.Б., Сивцов С.А. Средства электронной стабилизации телевизионных изображений для подвижных систем наблюдения. Тез. докл. 4-й конф. DSPA. – М.: МЦНТИ 2002. – С. 252-254.
4. Разработка информационной технологии многоцелевого обнаружения и сопровождения наземных движущихся объектов для бортовой системы обработки видеоизображений. Отчёт о научно-исследовательской работе № 26-03Г /№ госрегистрации 01200310355/ – Рязань, 2003. – 104 с.

**Д.В. ЧЕЛЯПОВА**

Рязанский государственный радиотехнический университет

### **3D-МОДЕЛИРОВАНИЕ КАМЕРЫ СОЛЯНОГО ТУМАНА**

*Рассматривается способ компьютерного моделирования работы камеры соляного тумана КСТ-0,4 с помощью создания трехмерных моделей в программе Blender и управления работой их в среде EmbarcaderoRadStudio 10.1 Berlin.*

Камеры соляного тумана – это оборудование, предназначенное для проведения климатических испытаний различных устройств и приборов в агрессивной среде тумана, в котором присутствует определенная концентрация соли. Крупнейшим Российским разработчиком таких камер является Волгоградская фирма «Спец Климат». Они производят различные модели камер для проведения климатических испытаний, одна из которых используется при проведении лабораторных работ по курсу «Методы и средства испытаний» в Рязанском государственном радиотехническом университете (Рис. 1).



**Рис. 1 – Камера соляного тумана КСТ-0,4.**

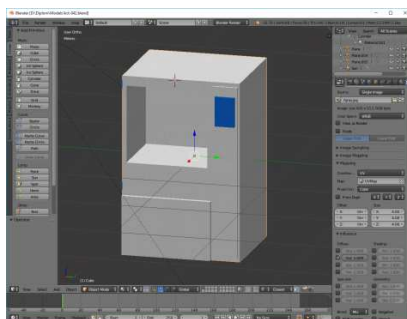
Технические характеристики этой камеры[1] показывают, что для её эксплуатации требуются:

1. Повышенное требование к мощности электропитания 380 В до 10 кВт.
2. Подвод холодной воды под давлением  $2\div 4 \text{ кгс/см}^2$ .
3. Подача сжатого воздуха под давлением  $1,93\div 3,92 \times 10^5 \text{ Па}$ .

Кроме этого, камеры имеют значительные габаритные размеры (1020×1190×1810 мм), вес (630 кг) и высокую стоимость (порядка 300 000 рублей).

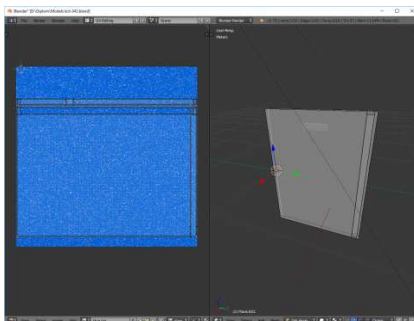
Также следует отметить, что технологический процесс испытания требует значительных временных затрат и приводит к выходу из строя испытуемых образцов. Все это позволяет сделать вывод, что проведения лабораторных работ практически невозможно осуществить из-за недостатка времени и высокой стоимости обслуживания камер соляного тумана.

Для решения этой проблемы было предложено разработать 3D-модели этой камеры и испытуемых приборов в программе Blender компании BlenderFoundation [2].



**Рис. 2 – 3D-модель корпуса камеры соляного тумана КСТ-0,4.**

Для предания большей схожести модели реальному прототипу использовалось UV-текстурирование отдельных элементов модели (Рис. 3).



**Рис. 3 – Текстурирование двери камеры соляного тумана КСТ-0,4.**

Аналогично были созданы 3D-модели испытуемого оборудования: вольтметра и амперметра (Рис. 4).



Рис. 4 – 3D-модели испытуемого оборудования.

Для управления этими модели было предложено использовать среду «EmbarcaderoRadStudio» [3]. В новой версии существует специальный компонент «Viewport3D», который позволяет размещать на форме 3D-модели (Рис. 5).

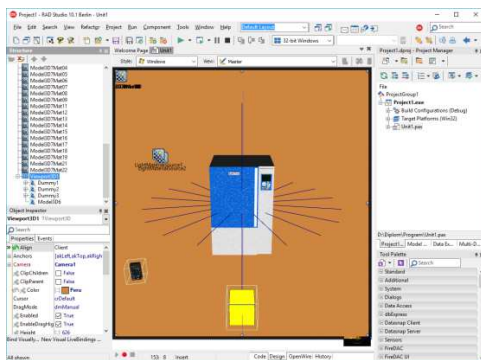


Рис. 5 – Размещение моделей в компоненте «Viewport3D».

Кроме моделей, которые были импортированы из программы «Blender» в формате «obj», на форме была размещена камера для просмотра и распределены источники света, для создания освещенности как с наружу, так и внутри камеры.

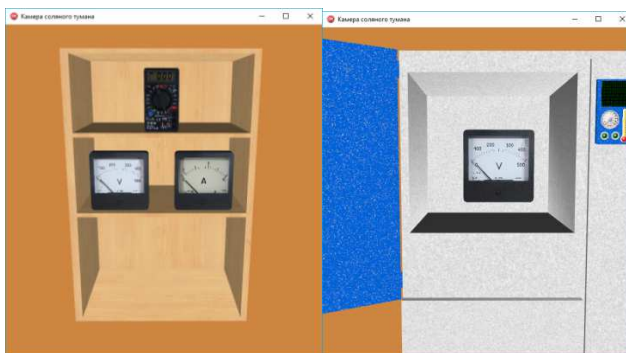
Для создания анимационного эффекта открытия двери (Рис. 6) к её модели был добавлен компонент «FloatAnimation».





**Рис. 6 – Анимация открытия двери камеры солевого тумана.**

После открытия двери пользователь должен выбрать испытуемый прибор и разместить его внутри камеры (Рис. 7).



**Рис. 7 – Выбор и размещение испытуемого прибора.**

Для моделирования работы камеры на панели управления были размещены двухмерные компоненты из библиотеки «InstrumenLab» компании «MitovSoftware» (Рис. 8). Используя эти компоненты пользователь может:

1. Задать температуру внутри камеры.
2. Указать влажность.
3. Включить и отключить питание.
4. Начать или продолжить испытание.
5. Определить время испытания и результат.



**Рис. 8 – Панель управления камеры соляного тумана.**

Задав параметры, пользователь можно проследить влияние соляного тумана на образец в зависимости от температуры, влажности и длительности испытания. Графические эффекты, которые создаются с помощью специальных компонентов «BoxBlurEffect» или «SmoothMagnifyEffect», позволяют смоделировать эффект разрушения испытуемого образца в реальном времени.

Таким образом, разработанную 3D-модель камеры соляного тумана можно использовать для проведения лабораторных работ по курсу «Методы и средства испытаний» сократив временные и материальные затраты на эксплуатацию реального оборудования, создав эффект присутствия у учащихся и полностью смоделировав весь технологический цикл испытания.

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Камеры соляного тумана [Электронный ресурс] // Спецклимат, [2015]. URL: <http://specklimat.com/tuman>(дата обращения: 20.01.2017).
2. Основы Blender v.2.42a (второе издание) [Электронный ресурс] / сост.: Джеймса Кронистера, [2004–2006]. URL: [http://blender3d.org.ua/book/Blender\\_242](http://blender3d.org.ua/book/Blender_242)(дата обращения: 10.02.2017).
3. FireMonkey Quick Start Guide [Электронный ресурс] //Embarcadero [2017]. URL: [http://docwiki.embarcadero.com/RADStudio/Seattle/en/Category:FireMonkey\\_Quick\\_Start\\_Guide](http://docwiki.embarcadero.com/RADStudio/Seattle/en/Category:FireMonkey_Quick_Start_Guide)(дата обращения: 22.03.2017).

**В.С. ЧИКАЛКИН**

Рязанский государственный радиотехнический университет

## **АДАПТИВНЫЙ АЛГОРИТМ РАБОТЫ СИСТЕМЫ СИНХРОННОГО МНОЖЕСТВЕННОГО ДОСТУПА**

Рассматривается метод адаптации работы синхронного алгоритма множественного доступа.

Связь в системах с множественным доступом можно рассматривать в терминах теории массового обслуживания. Каждый узел имеет очередь пакетов, которые должны быть переданы, а канал с множественным доступом является обслуживающим прибором. В идеальном случае обслуживающий прибор должен рассматривать все ожидающие пакеты как стоящие в одной очереди, которую нужно обслужить, используя соответствующую дисциплину обслуживания очереди.

К сожалению, обслуживающий прибор не знает, какие узлы имеют пакеты; аналогично в узлах не известно о наличии пакетов в других узлах. Таким образом, интересная особенность проблемы состоит в том, что информация о состоянии очереди является распределенной. Одним из многих методов, которые были разработаны для решения этой проблемы является метод свободного доступа для всех, при котором из узлов обычно посылаются новые пакеты немедленно в надежде на то, что другие узлы передавать не будут.

Основная идея синхронного алгоритма состоит в том, что каждый узел, не имеющий задолженности, просто передает новый поступивший пакет в первом окне после момента прихода, рискуя, таким образом, попасть в случайные конфликты, но это приводит к очень маленькой задержке, если конфликты возникают редко. Таким образом, при синхронной системе ALOHA пакеты передаются почти немедленно, при этом иногда попадая в конфликты.

Когда в синхронной системе ALOHA возникает конфликт, в каждом узле, который передавал один из вступивших в конфликт пакетов, обнаруживается конфликт в конце окна, и он становится задолжником. Такие узлы пропускают некоторое случайное число окон перед повторной передачей.

Если повторные передачи узлов с задолженностью достаточно рандомизированы, то будет правдоподобной аппроксимация общего числа повторных передач и новых передач в заданном окне пуассоновской случайной величиной с некоторым параметром  $G > \lambda$ . При таком



тельно малой и новые поступающие пакеты приобрели бы дополнительную ненужную задержку (за счет того, сразу становились бы задолженными). В случае адаптивной ALOHA, однако,  $q_r$  может повышаться до 1, когда оцениваемая задолженность незначительна, так что новые поступающие пакеты задерживаются только тогда, когда оценка показывает, что система уже перегружена.

Алгоритм работает, производя вычисление оценки  $\hat{n}$  задолженности  $n$  в начале каждого окна. Каждый задолженный пакет при этом передается независимо с вероятностью

$$q_r(\hat{n}) = \min \{1, 1/\hat{n}\}.$$

При каждом  $k$  оценка задолженности в начале окна  $k+1$  обновляется исходя из оценки задолженности и сигнала обратной связи в окне  $k$  в соответствии с правилом

$$\hat{n}_{k+1} = \begin{cases} \max \{ \lambda, \hat{n}_k + \lambda - 1 \} \\ \hat{n}_k + \lambda + (e - 2)^{-1} \end{cases}.$$

### СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бертсекас Д., Галлагер Р., Сети передачи данных.
2. Назаров А.А., Одышев Ю.Д., Исследование сетей связи с протоколами «Адаптивная Алоха», 2000.

### А.А. ШИПИЛОВ

Рязанский государственный радиотехнический университет

### ОСОБЕННОСТИ ИЗГОТОВЛЕНИЯ ПЕЧАТНЫХ ПЛАТ С ИСПОЛЬЗОВАНИЕМ 3D ПРИНТЕРА

*Рассматривается применение 3D принтеров для создания печатных плат методом послойного наплавления.*

Современную жизнь невозможно представить без печатных плат. Для создания новых электронных приборов крайне важен быстрый и низкокзатратный процесс создания печатных плат. Но обычно на практике все обстоит иначе. Зачастую в разработке печатных плат, огромное значение имеет прототипирование или макетирование. С помощью этапа прототипирования можно быстро оценить работоспособность будущей печатной платы. И зачастую у радиолюбителей и

небольших предприятий нет возможности быстро и качественно создать прототип, приходится заказывать изготовление плат «на стороне», ожидая недели, а порой и месяцы. А в случае, если потребуется внести в плату даже малейшее изменение, этот долгий и затратный процесс нужно начинать сначала.[1]

Решением данной проблемы может служить 3D принтер. Он способен сократить процесс создания печатной платы с нескольких месяцев до нескольких часов.

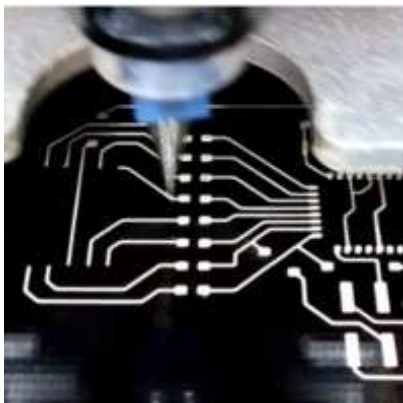
3D-печать, однажды, серьезно изменила представление о скорости производства разных устройств и их составных элементов, и теперь длительный процесс создания печатных плат выбивается из общей картины современного прогресса.[4] К тому же, большие расходы на изготовление прототипов новых электронных приборов (этап прототипирования аппаратуры), могут только усугубить ситуацию с уже существующей нехваткой инженеров в России. Подобные трудности в производстве печатных плат могут негативно сказываться на производительности предприятий, которые относятся к малому и среднему бизнесу, и в особенности на работе стартапов. Вместе с тем, долгий процесс создания плат неизбежно приводит к задержке выпуска на рынок новых технологий, что крайне неблагоприятно сказывается на развитии технического потенциала страны.

В настоящее время можно выделить несколько направлений по созданию печатных плат с использованием 3D принтера:

- удаление ненужного проводящего слоя (или попросту фрезеровка медного слоя);
- удаление защитного слоя с основания печатной платы (та же фрезеровка только уже нанесенного защитного слоя с последующим травлением и удалением защитной краски);
- лазерное гравирование (наносится защитная краска и лазер выжигая, удаляет ненужные участки, также после необходимы этапы травления и удаления краски);
- нанесение защитного слоя на основание печатной платы (то есть проводящие дорожки изолируются друг от друга с помощью защитной краски, также с последующим травлением и удалением краски);
- печать токопроводящих дорожек на поверхности разных материалов (основан на методе послыонного наплавления материала);
- заполнение полостей токопроводящим материалом [3].

Я считаю, что наиболее перспективным будет являться 3D принтер, который осуществляет непосредственную печать токопроводящих дорожек на поверхности разных материалов.

На рис. 1 изображено нанесение расплавленного металла на стеклотекстолит.



**Рис. 1 – Нанесение расплавленного металла на стеклотекстолит.**

Суть работы такого принтера заключается в следующем. Готовый gerber файл печатной платы конвертируется в g-code и загружается в принтер. Далее принтер наносит токопроводящие чернила на закрепленную на столе подложку. После стол 3D принтера нагревается до температуры отвердевания клея, который входит в состав чернил. Это необходимо чтобы чернила приклеились к основанию печатной платы.[2]

Следующим этапом необходимо нанести на контактные площадки припой, расставить элементы и снова нагреть стол 3D принтера до температуры плавления паяльной пасты. Далее плата должна остыть и она готова к применению.

Если сравнить данную разработку и популярные в последнее время «бредборды», то изготовление печатных плат на принтере даст инженеру больше свободы. Не стоит забывать про самостоятельное травление печатной платы. Этот процесс достаточно недорогой, но сложный и требует много времени. Целью же является как раз сделать процесс изготовления печатных плат более простым, не допуская значительного увеличения их себестоимости.

3D технологии – это технологии будущего. 3D принтеры могут стать отличным дополнением для производства в целом.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Design for 3D Printing: Scanning, Creating, Editing, Remixing, and Making in Three Dimensions. The Editors of MAKE. 2015.

2. Voltera V-One: быстрое изготовление печатных плат для прототипов электронных устройств [Электронный ресурс], URL: <https://geektimes.ru/company/dadget/blog/246644> (дата обращения 11.12.16).

3. 3D-печать в радиоэлектронике [Электронный ресурс], URL: <http://3dtoday.ru/blogs/top3dshop/3d-printing-in-electronics> (дата обращения 15.01.17).

4. 3D-принтер – технологии, использование, как работает [Электронный ресурс], URL: <http://3d-week.ru/3d-printer> (дата обращения 14.02.17).



Уважаемые коллеги!

Министерство образования и науки Российской Федерации, Российский фонд фундаментальных исследований и Рязанский государственный радиотехнический университет проводят **15-17 ноября 2017 года XXII Всероссийскую научно-техническую конференцию студентов, молодых ученых и специалистов «Новые информационные технологии в научных исследованиях» (НИТ-2017)**.

Работа конференции будет проходить по следующим направлениям:

- *информационные технологии в социальных и экономических системах;*
- *математические модели в информационных технологиях;*
- *телекоммуникационные технологии;*
- *информационные ресурсы и программно-инструментальные средства;*
- *технологии искусственного интеллекта в проектировании;*
- *информационные системы и процессы;*
- *обработка изображений в системах управления;*
- *геоинформационные технологии;*
- *информационные технологии в промышленности;*
- *космические информационные технологии;*
- *автоматизация проектирования в телекоммуникационных системах.*

Для участия в конференции необходимо:

- **до 1 июля 2017 г.** по эл. почте ([nich@rsreu.ru](mailto:nich@rsreu.ru), тема НИТ-2017) выслать заявку на участие с названием доклада.
- **до 1 сентября 2017 г.** по эл. почте ([nich@rsreu.ru](mailto:nich@rsreu.ru), тема НИТ-2017) выслать тезисы доклада (имя файла по фамилии автора).
- **до 1 октября 2017 г.** Оплатить регистрационный взнос (300руб).

**Регистрационный взнос** включает расходы на издание программы и трудов конференции. Перечислить оргвзнос можно на карту ОАО «Сбербанк России»: \_\_\_\_\_ или оплатить лично: аудитория 126а (кафедра САПР ВС) Чернаковой Наталье Викторовне. Телефон для справок: (4912) 46-03-09 – Периго Наталья Борисовна.

Приглашения на конференцию будут высланы по указанным авторами адресам. Сборник тезисов будет опубликован к началу конференции. **ЭЛЕКТРОННАЯ ВЕРСИЯ СБОРНИКА БУДЕТ РАЗМЩЕНА В E-library И ПРОИНДЕКСИРОВАНА В РИНЦ.**

Подробная информация – на странице в сети Интернет:  
[nit.rsreu.ru](http://nit.rsreu.ru)

### **Требования к оформлению**

Тезисы доклада объемом 1-2 страницы должны быть представлены в виде документа в формате текстового процессора Microsoft Word. Размер листа – А4, поля со всех сторон 20 мм. Шрифт Verdana. Размер шрифта – 12 кегля, текст печатается через 1 интервал и должен быть выровнен по правой и левой сторонам документа. Красная строка 0.5 см. Автоматическая расстановка переносов. Формулы подготавливаются с применением Microsoft Education 3.0, рисунки в формате BMP или JPEG с разрешением 200x200 dpi, подрисовочная подпись не должна быть частью рисунка. Количество формул и рисунков должно быть минимально возможным.

Тезисы доклада должны быть оформлены в следующем порядке: название доклада прописными буквами по центру, шрифт - полужирный; следующая строка – авторы (инициалы перед фамилией!) по центру; следующая строка – научный руководитель (Ф.И.О., ученая степень, ученое звание) по центру; следующая строка – название организации по центру; далее через одну строку текст доклада. Библиографический список (если имеется) – по ГОСТ 7.1.2003 и отделяется от текста пустой строкой. Ссылки на библиографический список в тексте доклада в квадратных скобках.

### **Образец оформления ВОССТАНОВЛЕНИЕ НЕСУЩЕЙ СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ**

И.И.Иванов

Научный руководитель – Петров П.П.,  
д-р техн. наук, профессор

**Рязанский государственный радиотехнический  
университет**

В докладе рассматривается ряд проблемных вопросов по проектированию...

#### *Библиографический список*

1. Сосулин Ю.Г. Теоретические основы радиолокации и радионавигации: учеб. Пособие для вузов. – М.: Радио и связь, 1992. – 304 с.

# **Информационные технологии**

Межвузовский сборник научных трудов

Компьютерная верстка: А.Н. Сапрыкин,  
А.В. Соколов, К.Ю. Филатов.

Подписано в печать 15.05.17. Формат бумаги 60x84 1/16.  
Бумага офсетная. Печать струйная. Усл. печ. л. 6,4.  
Тираж 500 экз. Заказ № 902.

Отпечатано в типографии BookJet

390046, г. Рязань, Скорбященский проезд, д.20,  
тел.:+7 (4912) 466-151, сайт <http://bookjet.ru/>